PROJECTS ■ THEORY ■ APPLICATIONS ■ CIRCUITS ■ TECHNOLOGY

NUTS AND VOLTS

www.nutsvolts.com
December 2016

EVERYTHING FOR ELECTRONICS

A DIGITAL ANALOG

CAN A PIC REPLACE A 555?

TAKE THE CAN BUS for a spin

EXPLORING NEAR SPACE with multispectral imaging

BUILD THE AUTOMATIC sump pump controller

SPECTRUM SHORTAGE a threat to future wireless

Expert Engineering

Manufactured with hitecnology

Since 1983, Hitec has designed, developed and manufactured servo technologies that push the boundaries of imagination and innovation. Consider, for example, our EMI-shielded case, giant scale duo: the HS-1005SGT, with its specially-handmade cored motor and steel gears or our HS-1100WP, with robust torque and an IP-67 waterproof rating. Their unrivaled reliability, strength and durability deliver expert performance for your most challenging and demanding industrial and commercial applications.

	HS-1005SGT
Amplifier Type	High Response Digital MOSFET
Motor Type	5 Pole Cored Carbon Brush
Operating Voltage Range	11.1 ~ 14.8 Volts DC
No Load Speed Range	0.19 - 0.26 Sec @ 60°
Peak Torque Range	84.0 - 110.0 kg-cm 1166 - 1528 oz-in
Dimensions	64 x 33 x 73mm 2.52 x 1.3 x 2.90in
Weight	363g / 12.8oz
Gear Type	Hardened Steel with MP First Gear
Bearings	2 Ball Bearings and 2 Needle Bearings
Output Shaft	10mm 15 tooth

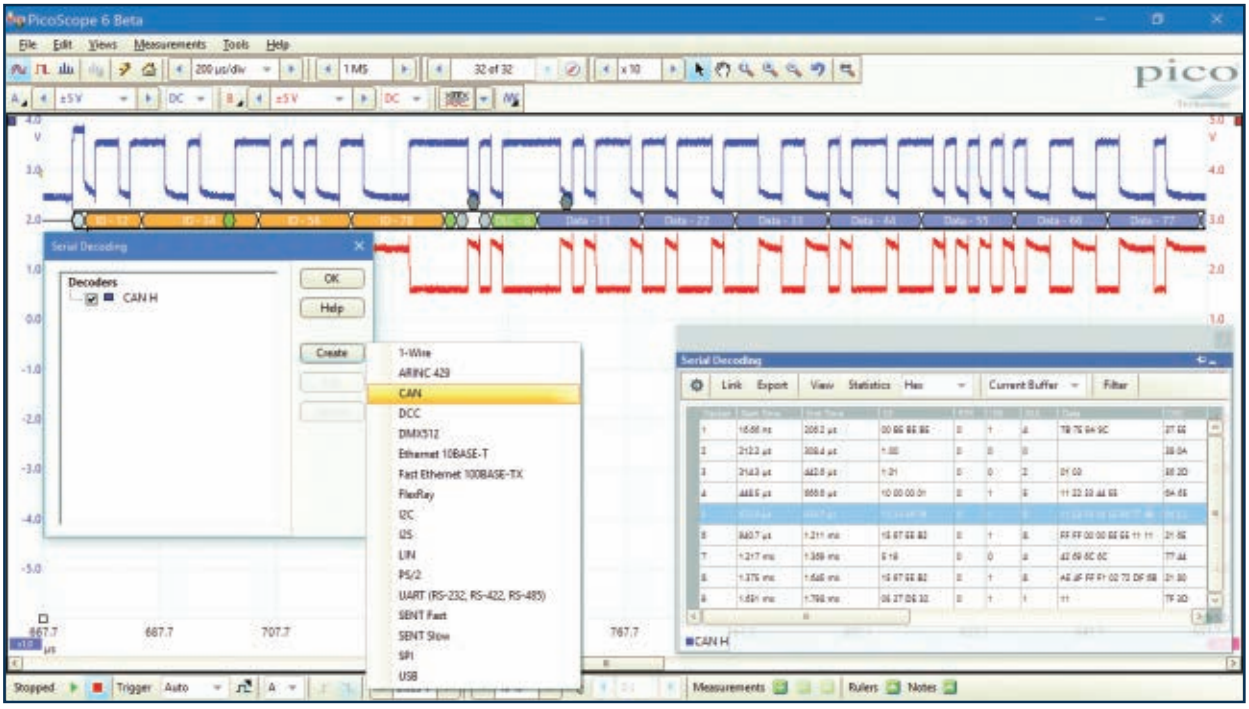
	HS-1100WP
Operating Voltage Range	11.1 ~ 14.8 Volts DC
No Load Speed Range	0.19 - 0.26 Sec @ 60°
Peak Torque Range	84.0 - 110.0 kg-cm 1166 - 1528 oz-in
Maximum Current Draw	5500 - 6500mA
Dimensions	64 x 33 x 73mm 2.52 x 1.3 x 2.90in
Weight	363g / 12.8oz
Gear Type	Hardened Steel with MK™ First Gear
Output Shaft	6.5mm Square

Custom design and manufacturing available.

HITEC RCD USA, Inc. | 12115 Paine St. | Poway, CA 92064 | [858] 748-6948 | www.hitecrcd.com

WAVEFORM ENGINEERING ON YOUR DESKTOP

PicoScope PC oscilloscopes deliver the functionality of six traditional instruments in an all-inclusive package – at a price you can afford.



Two-channel, four-channel and mixed-signal models have the functionality of an oscilloscope plus a Logic Analyzer (on MSO models), Spectrum Analyzer, Function Generator, Arbitrary Waveform Generator and Serial Bus Analyzer with support for 16 protocols included as standard.

Flexible resolution models enable measurement of high-speed signals with timing resolution to 1 ns, and analysis of low-frequency waveforms with 16-bit resolution and > 70dB SFDR.

In addition to the PicoScope 6 desktop application, the Software Development Kit (SDK) can be used to write custom programs in scientific, research and electronics applications.

For more information please visit www.picotech.com or call 1-800-591-2796

Email: sales@picotech.com. Errors and omissions excepted. Prices are correct at the time of publication. Sales taxes not included. Please contact Pico Technology for the latest prices before ordering.

December 2016

Subscription Information
Nuts & Volts — PO Box 15277
North Hollywood, CA 91615-9218
Call 877-525-2539 or go to www.nutsvolts.com
Subscribe · Gift · Renewal · Change of Info

22 Build a Sump-Fill Level Controller
Whether you need to keep track of high or low fluid levels, these two versions of a level monitor will do the job without any moving parts, linkages, floats, or switches that can stick or fail.
■ *By Ron Hoffman*

29 Take a CAN Bus for a Spin
A Controller Area Network (CAN) bus is designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. This three-part series will give you the knowledge you need to utilize this protocol. First up: Learn how to use a CAN bus with a Propeller MCU.
■ *By Jon Titus KZ1G*

34 Replacing the 555 with a PIC
Inspired by Forrest Mims and his *Mini-Notebook* series, we will detail the emulation of a 555 or 556 using a PIC in several different circuits in our own series. The particular implementation this month covers both monostable and astable modes, and the PIC does not have to be reprogrammed in any way.
■ *By Larry Cicchinelli*

40 Computer Control and Interfacing with the NI MyRIO
National Instruments' MyRIO is one of the best units available for computer interfacing and data acquisition. It places dual-core ARM Cortex-A9 real time processing and Xilinx FPGA customizable I/O into the hands of novices. Get an introduction to MyRIO and deploy your first LabVIEW virtual instrument as we begin a five-part tutorial.
■ *By David Ward*

38 *Name That Part!*
Think you know your stuff? Try this final photo quiz of 2016 that will test your memory and your wits.
■ *By David Goodsell*

08 Q&A
Reader Questions Answered Here
Topics covered this month include VCO devices and a design; padding 70 volt audio; and dealing with an ICOM antenna tuner problem.

12 Open Communication
The Latest in Networking and Wireless Technologies
Spectrum Shortage Threatens the Future of Wireless
There is only so much spectrum to be had. Fortunately, technology and regulation are stepping up to meet the growing demand for space in the already crowded airwaves.

16 Near Space
Approaching the Final Frontier
Cheap Multispectral Imaging for Amateur Science
Satellites and airplanes are the primary platforms in use today for imaging Earth's surface. However, satellites are very high above the ground and expensive, while airplanes are unable to safely view the ground up close. This means drones like quadcopters are the solution to Earth imaging very close to the ground and BalloonSats are the cheap alternative to satellite-based imaging.

48 Practical 3D Printing
Real World Uses for Electronics Experimenters
3D Printed On-Air Flashing Sign
Don't you just hate it when you're recording a video to put up on YouTube and someone walks in on you in the middle, causing you to have to start over? Well, then we have the sign to give people for you!

52 The Design Cycle
Advanced Techniques for Design Engineers
Exploring the Nuances of Nordic Semiconductor's nRF52832's Timer and GPIO Peripherals
The Nordic Semiconductor nRF52832 is a Bluetooth radio that just happens to be supported by an awesome ARM CPU. In this installment, we're going to concentrate on exploring the nuances of the nRF52832's Timer and GPIO peripherals.

Departments

06 DEVELOPING PERSPECTIVES
Spy Gear
07 READER FEEDBACK
20 NEW PRODUCTS
21 SHOWCASE
58 CLASSIFIEDS
60 NV WEBSTORE
62 TECH FORUM
64 ELECTRO-NET
64 AD INDEX

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree #40702530) is published monthly for $26.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to **Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@nutsvolts.com.

2-For-1 Holiday Offer

Give a one year *(12 issues)* subscription to a friend and receive a *FREE* one year subscription for another friend, *or yourself!*

Not only will they get a full year of Nuts & Volts delivered right to their door, they'll also get free access to our digital (web) edition, PLUS our mobile (iOS, Android, Kindle Fire) edition.

Subscribe online at:

www.nutsvolts.com

or call:

877-525-2539 (toll free)

818-487-4545 (outside US)

Be sure to use promotion code Y6WXMSA when ordering

U.S. Price: **1 Yr - $26.95**

DEVELOPING PERSPECTIVES

by Bryan Bergeron, Editor

Spy Gear

When I was a kid, the coolest electronics were miniature wireless microphones, video cameras, and other gear that could be stuffed into a ball point pen, the heel of a shoe, or a clock radio. Today, that sort of spy gear is mainstream. It most often involves home monitors, remote two-way video communications, and similar technology used to "spy" on the sitter, the kids, or elderly parents, or even the pet — all from the comfort of a smartphone.

Remote, audio, and video monitoring is much more than just cool technology. The convenience it provides is close to addicting. I can't imagine going back to a time when I had to physically walk to the front door and peek through the peephole to see who was knocking. Or of wondering what the kids are doing in the back yard while I'm working on a project in my workshop.

Of course, when I was a kid, I never dreamed of having live streaming video available from hundreds of feet in the air. The closest I ever came to a quadcopter with video camera was a glider with a built-in film camera with a timer. If the glider happened to be directly overhead when the timer went off, I'd have a print of an aerial photo a week to 10 days later.

I've built several quadcopters over the past few years, and I'm still hooked on the aerial images. It doesn't help my bank account since miniature streaming video cameras keep getting lighter and higher resolution every few months.

I used to have one of those spy microphones made of a handheld microphone and clear plastic parabolic reflector. I think they were sold as bird song recording microphones to get around the privacy issues. In any case, they worked. Today, however, highly directional amplified microphones can be found in any home video shop. Still, it's a wonder what you can do with a plastic salad bowl, a directional microphone, and a good preamplifier.

Of course, the real spy gear today is necessarily secret. I've heard of using lasers reflected off of window panes that vibrate in response to the pressure waves associated with voice. Then, there's the use of microwave radiation to pick up vocal movement from those in the path of the beam. There's also the flying bird camera that looks like a wounded bird. And don't forget the multi-million dollar imaging satellites that monitor activity on the surface.

I did a quick search for "spy gear" on Amazon, which turned up 6,020 items ranging from lie detectors, night vision goggles, walkie-talkies, and voice changers, to quadcopters and snake cameras. Even more impressive is that most of these spy devices are in the $20-$30 range. True, they're toys, but the technology is still a bargain. Plus, there's no better way to understand a voice changer or any other spy electronics than by performing a non-destructive teardown.

Modern "toy" spy technology touches on just about every aspect of electronics, from sensors and communications to microcontrollers. As such, spy technology is a fun vehicle for learning basic electronics. Even if you have your mind set on $2,500 army surplus night vision goggles, starting off with a $40 version sold in the toy store is probably the best way to go. You might be surprised at how little difference there is between the toy and professional versions of the technology.

So, go on. Order yourself a few electronic spy toys from Amazon or other retailer. It's fun viewing electronics technology from a fresh perspective. **NV**

NUTS AND VOLTS

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct.
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore orders only **1-800-783-4624**
www.nutsvolts.com

Subscription Orders
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

FOUNDER
Jack Lemieux

PUBLISHER
Larry Lemieux
publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ ADVERTISING SALES
Robin Lemieux
robin@nutsvolts.com

EDITOR
Bryan Bergeron
techedit-nutsvolts@yahoo.com

VP OF OPERATIONS
Vern Graner
vern@nutsvolts.com

CONTRIBUTING EDITORS
Fred Eady
Lou Frenzel
Paul Verhage
Jon Titus
David Ward
Kristen McIntyre
Chuck Hellebuyck
David Goodsell
Larry Cicchinelli
Ron Hoffman

CIRCULATION DEPARTMENT
subscribe@nutsvolts.com

SHOW COORDINATOR
Audrey Lemieux

WEBSTORE MARKETING COVER GRAPHICS
Brian Kirkpatrick
sales@nutsvolts.com

WEBSTORE MANAGER/ PRODUCTION
Sean Lemieux
sean@nutsvolts.com

ADMINISTRATIVE STAFF
Re Gandara

Copyright © 2016 by T & L Publications, Inc.
All Rights Reserved
All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879**.

READER FEEDBACK

Clock Talk

Bill van Dijk's September 2016 article, "Build the Numitron, A Six-Digit Clock," is just plain fabulous! I really enjoyed it, and what's more, I have learned something new, and that is important to me. I now have several questions for Bill.

For soldering surface-mount components, you mention that your fine point tip for your soldering iron is almost needle sharp. It looks like our soldering stations might be the same. I have a digital Weller WESD51 powering a PES51 iron. I bought the model "ETH" tip, but have always thought I needed a finer point for my SMD exploits. Have you found a finer tip to fit onto your iron other than the Weller ETH?

When I built a PIC16F877 based clock a few years ago, I used a single 4 MHz crystal. The interrupt routine and Timer1 perform timekeeping using that frequency. In my climate controlled house, I routinely see accuracy better than 10 seconds per year. In your design, you added a second crystal with capacitors to do the timekeeping at a slower frequency. Why? Is there an advantage in using the additional parts?

I really admire your use of a hardware solution for debouncing your pushbutton switches. The link you included for the debouncing discussion (**www.eng.utah.edu/~cs5780/debouncing.pdf**) is fantastic. I have never seen a more thorough handling of the topic. However, I believe that author does make one mistake: As voltage climbs, the high-going transition point for a Schmitt trigger is not 0.9 volts, so I think some of his results are in error. It seems that he has mixed up the two transition point voltages.

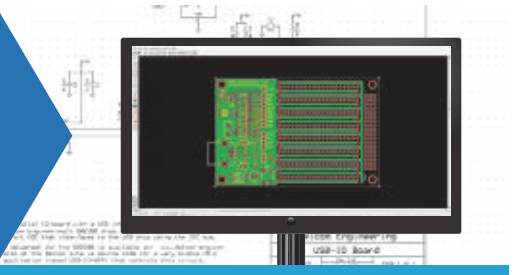
After reading that in-depth debouncing discussion and then looking again at your schematic, I became totally befuddled. You do not seem to follow that author's logic in your circuit. Your 15K discharge resistors (R19 and R20) are very similar in value to those employed in the discussion (R2 is 18K), so at first glance it appeared you were on the same track. However, your debounce capacitors are specified as 0.1 µF instead of 1.0 µF, which would certainly be a big change in the math equations. Plus, your charge resistors (R17 and R18) are only 1K compared to the discussion's R1 of 101K. Coupled with the diodes D3 and D4 in your debounce circuit, that is a very fast charge time indeed! Could you tell me how you decided to use the 1K resistor here?

I used the formulas in the discussion to do the math for your debouncing circuit. The port C pins you use on your PIC are Schmitt triggers, so the threshold voltages are one volt low-going and four volts high-going. With your components, that will take care of bouncing contacts when a button is pressed for 2.4 milliseconds. In the discussion, it was decided that a good target would be 10 to 20 milliseconds. I bet you have good quality pushbutton switches in your clock, and with reduced bouncing in quality contacts I imagine your functionality does not suffer. If another builder puts this circuit together with old or inferior switches, unfortunately, there could be contact bounce problems.

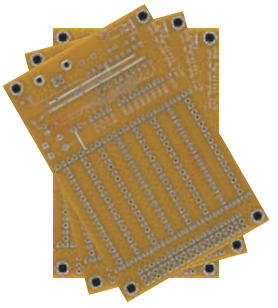
I am more interested in the button release, however. I did the math, keeping in mind the bypass diode. When you release your pushbutton, that 1K resistor with the 0.1 µF cap will only suppress bouncing for 0.15 milliseconds. I understand that released contacts do have less bounce than the press-down, but some bounce hash is present in

Continued on page 58

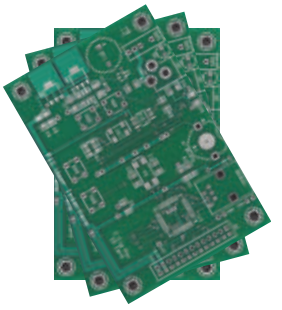
FREE
PCB Layout Software &
PCB Schematic Software



expresspcb.com



3 PCBs
$41 + shipping
2-layer
MiniBoard
(Shipped in 1 business day)



3 PCBs
$61 + shipping
2-layer
MiniBoardPro
(Shipped in 2 business days)

3 PCBs
$81 + shipping
4-layer
MiniBoardPro
(Shipped in 3 business days)

1 DOWNLOAD our free CAD software
2 DESIGN your 2 or 4-layer PCB
3 CHECK your design with xCHECK DRC
4 SEND us your design with just a click
5 RECEIVE top quality boards in just days

Now Offering
SMT
STENCILS
– Nano Coating available–

MADE IN THE USA | www.expresspcb.com

In this column, Kristen answers questions about all aspects of electronics, including computers, circuits, electronic theory, troubleshooting, radio, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. **Send all questions and comments to: Q&A@nutsvolts.com.**

- **VCO Devices and a Design**
- **Padding 70 Volt Audio**
- **ICOM Antenna Tuner Problem**

VCO Devices and a Design

Q I'm trying to build a VCO and found a useful diagram that shows a LM566 as the heart of it. Unfortunately, that part is discontinued. I have found other example circuits, but I was hoping to find something closer to the LM566 as the default design just seems cleaner to me. Can you recommend a part that might be similar?

Cari Bryant
Chicago, IL

A There are many different VCO designs that are possible. The LM566 is very nice in that it has outputs that give you both triangle waves and square waves. There is a newer device from Texas Instruments: the SN74LS624. You can find it on the TI website at **www.ti.com/product/sn74ls624**. It seems to be nicely linear and thermally stable. Sadly, it doesn't give function generator style waveforms, but you can create the other waveforms if you need them.

Still, though, what if that device wasn't available? What's a simple oscillator that could be voltage controlled? A very simple oscillator can be made using an op-amp in a non-linear way. We exploit the concept of hysteresis: the lagging property of a system. This kind of oscillator is sometimes called a relaxation oscillator.

An op-amp can be used as a comparator so that its output is either plus or minus, depending on whether the + input is above or below the - input. So, if the + input is above the - input, the output of the op-amp will swing to somewhere near its positive voltage rail. Vice versa if the roles of + and - are reversed.

As you might recall from last month's column, we discussed the relationship between voltage and current in a capacitor (**Figure 1**). That relationship can be used to calculate (solving a differential equation) the voltage on a capacitor when the current is supplied through a resistor. It is the classic exponential curve as described in **Figures 2a** and **2b**. If we have some voltage across that series R-C, it will rise according to that curve, with a time constant that is proportional to the values of R and C and the voltage applied. We will apply that voltage with an op-amp's output. Since it rises over time, we can use it as a timing element.

$$I = C\frac{dV}{dt}$$

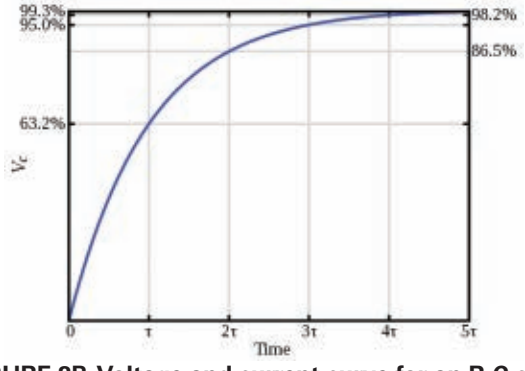
■ **FIGURE 1. Voltage and current relationship for a capacitor.**

The trick to get it to oscillate is to monitor the voltage across the capacitor with one of the op-amp's inputs, and then have the op-amp output switch the other way when it crosses some threshold. We want the output voltage to be the opposite of the capacitor's voltage in order to make the circuit reverse its function and oscillate, so we use the - input. This creates a phase reversal that's needed for oscillation. So, to create a simple oscillator at a reasonable and predictable frequency, we introduce some of that hysteresis (or lag) by changing the threshold we present at the + input, depending on the state of the oscillator. We do this with a voltage divider that gives a fraction of the op-amp output voltage as the threshold, and the period is dependent on the voltage rise of the R-C series circuit (**Figure 3**).

We can get square waves from the op-amp's output, and exponential curves from the capacitor voltage. If we want to use the capacitor voltage for some purpose, it would be a good idea to buffer it with, say, another op-amp since changing the current at that node will change

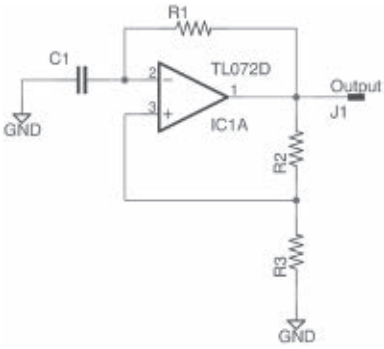
$$V(t) = V_0 e^{-\frac{t}{RC}}$$

■ **FIGURE 2A. Voltage and current relationship for an R-C circuit.**

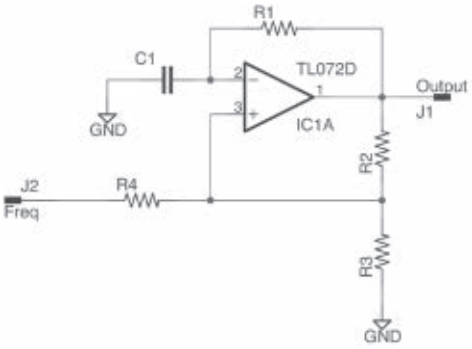


■ **FIGURE 2B. Voltage and current curve for an R-C circuit.**

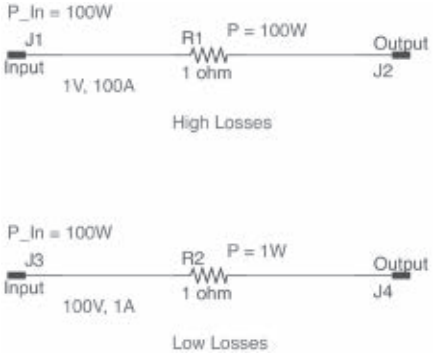
Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/magazine/article/December2016_QA.



■ **FIGURE 3. Simple relaxation oscillator.**



■ **FIGURE 4. Relaxation oscillator with frequency control.**



■ **FIGURE 5. Loss for high impedance vs. low impedance.**

the oscillating frequency. Triangle waves would take a bit more work, but can be done with an additional op-amp. I'll leave that as an exercise for the reader, but think about last month's column.

Now, how do we make this frequency controlled? There are a few different ways; for example, with a varactor diode that acts as a voltage-dependent capacitor or by creating a bidirectional current source to change the threshold voltage. For this circuit, though, we will just do something simple to show the principle.

We can use a resistor to yank on the voltage divider node on the + input to the op-amp, and that will change the threshold, making the duty cycle asymmetric in the process but also changing the frequency. The change in threshold will change where on the charge curve of the capacitor that the op-amp output flips. The frequency control input on the circuit shown in **Figure 4** will definitely need some current, so it should be buffered if it's driven by anything other than a low impedance source.

Also, the output frequency will be rather non-linear with voltage. To fix that, we'd need to do something more complicated that uses current sources to both linearize the capacitor charging rate and to linearize the change in threshold with voltage. A varactor diode won't be linear either, except in a certain region. In some future column, we can explore how to make a better VCO.

Padding 70 Volt Audio

Q I have been tasked with extending the sound system in our meeting hall for our local community center. One of the requirements is to be able to adjust the volume of a few new speakers that are in some of the adjoining rooms. I had planned to use L-Pads, but the twist here is that the system uses 70 volt transformers. I am at a bit of a loss as to where to place these L-Pads. Do they go between the 70 volt line and the transformer, or between the secondary of the transformer and the speaker?

Avis R. Ames
Fayetteville, NC

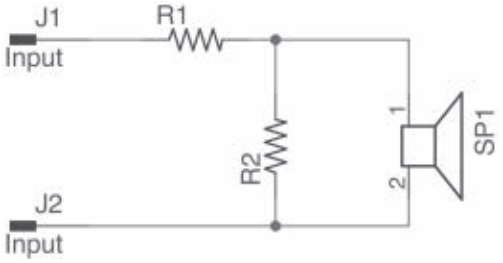
A Seventy volt audio systems are designed to work in a way similar to the power company's high voltage lines. The voltage on the wires is increased so that the resistance of the wire contributes a smaller overall voltage drop. The current is lower for the same power: P = VI. So, if we increase V, I is decreased for a given value of P. See **Figure 5** for an example that illustrates this point.

Since audio signals are AC, a transformer can be used to transform the impedance — much like the power company — to convert the high voltage/low current input to a low voltage/high current signal that a four or eight ohm speaker expects.

Now, padding down the audio means lowering the voltage presented to the speaker system while roughly maintaining the system's expected impedance. It's called an L-Pad because the resistor network is in an L configuration with one resistor in series and another in parallel with the speaker (**Figure 6**). Those resistors are typically variable (i.e., potentiometers) and the potentiometers are ganged and matched such that the impedance doesn't vary much while the power delivered to the speaker can be lowered.

Whether this is done on the high impedance side or done on the low impedance side is not that important electrically, except that the required resistances will be different. What is probably more important is that the L-Pad is able to be placed where you need it to be.

Chances are that the L-Pad would be easier to put somewhere in the distribution line, far from the speaker; perhaps mounted in a wall. That would argue for a high



■ **FIGURE 6. Speaker L-Pad example.**

impedance L-Pad on the 70 volt side. It turns out that high voltage L-Pads are readily available. I found a couple on Amazon (the Bogen AT10A and AT-35A) made to control 10 and 35 watt systems at 25V or 70V. This may be just what you want rather than building them yourself, as it might be hard to find the right potentiometers. Of course, we could build that if we wanted to.

ICOM Antenna Tuner Problem

Q I own an ICOM antenna tuner labeled IC-AT500 which doesn't work all the time. I cannot find a service manual, and there has been NOTHING on the Internet about who, where, or what can handle this product. My problem is during transmission when yaking into the mic, the antenna tuner will indiscriminately change bands WHILE I AM TRANSMITTING.

For example, I will be transmitting on 20 meters and the tuner will simply switch to 15 meters or 10 meters, or in some cases, display BOTH LEDs on the front chassis. This has only recently occurred within the last four years, and I have owned the tuner for over 25 years.

73

Dennis Marandos K1LGQ

A For all of you other ham radio operators out there, I'm happy to answer ham radio questions too. Since not everyone is a ham, we'll keep things general, but some problems and explanations are particular to ham radio. As a side note, being a ham radio operator is really fun and educational, so maybe think about getting licensed. Information about becoming a ham can be found at the ARRL website at **arrl.org**.

I did find the English manual on the ICOM website at **www.icom.co.jp/world/support/download/manual/pdf/IC-AT500_AT100.pdf**. It doesn't talk about such problems, though.

For those not familiar, an antenna tuner is something similar to the L-Pad described in the last question, except that it's frequency specific and not used for attenuation. It uses reactive elements (inductors and capacitors) to match an antenna's impedance to what a transmitter expects and is happy with.

Most modern amateur transmitters expect a 50 ohm resistive load impedance and can't tolerate a load much outside of that. Usually, they begin to limit their power output at around a 1.5:1 standing wave ratio. The tuner is typically an L or T network that transforms a reactive and resistive impedance at the antenna feed point to

The Easiest Way to Design Custom
Front Panels & Enclosures

You design it
to your specifications using our FREE CAD software, Front Panel Designer

We machine it
and ship to you a professionally finished product, no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

FRONT PANEL EXPRESS
FrontPanelExpress.com

AP CIRCUITS
PCB Fabrication Since 1984

As low as...
$9.95 each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com

VISA MasterCard PayPal IPC MEMBER ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES BBB

the desired 50 ohms resistive. It can do this within limits, depending on the range and power handling capabilities of the capacitors and inductors. There are also some theoretical considerations to the matching range.

The problem you describe sounds like some kind of RF incursion, though it's hard to exactly diagnose it from afar. If the impedance being matched is far from the system impedance of 50 ohms, the voltages or currents inside the tuner can get quite high. This, in turn, causes there to be strong electric and magnetic fields which can induce voltages and currents in parts of the tuner that don't expect it.

In addition, if there are high RF fields in your shack (where the radio is for you non-hams), those fields can induce variations in the power supply voltage and current to the tuner, as well as actually signal the tuner to switch bands through its band-switch control inputs.

A trick that we can use to minimize those effects is to add inductance to the power supply lines — and perhaps elsewhere inside the tuner — to limit the induction of these unwanted voltages and currents. There are at least two things to employ. The first is to use clip-on ferrites to increase the inductance of the power supply wires. Try to get the biggest ones you can find, and wrap the wires around the ferrite as many times as you can.

Since this particular tuner is powered off the AC mains, it may be difficult to get more than one turn with a thick power cord, but try as best you can. If you are using band control wires to the rig (the transceiver), give them the same treatment. If you can find out what the ferrite mix is, try to get one that provides a large relative μ, the magnetic permeability when compared to a vacuum, at or below 30 MHz. That will give you the highest inductance. I usually put those ferrites everywhere I can think of. They're typically pretty cheap.

The other thing is to try to either shield or AC short those band switch connectors (labeled Accessory Sockets) on the back, if you aren't using them. It's mechanically hard, but putting a metal cap over them would provide some shielding. The better thing might be to attach a capacitor from the band switching control voltage input (pin 13 according to the manual) to the chassis. Perhaps a ceramic disc capacitor around 0.01 μF would do the trick.

Give these tips a try and write back to the Q&A column to let me know how you fared. **NV**

Have a question for Kristen?
Email it to Q&A@nutsvolts.com
Comments are also welcome!

■ BY LOU FRENZEL W5LEF

Spectrum Shortage Threatens the Future of Wireless

Technology and Regulation to the Rescue

Wireless continues to be the fastest growing segment of electronics. Its growth is driven by cellular data, Wi-Fi, and other short-range technologies. The Internet of Things (IoT) and the related machine-to-machine (M2M) movements are also creating massive growth opportunities for wireless devices. The push for faster and expanded cellular services is driven by the increased use of streaming video from Netflix, YouTube, and others.

What all these things have in common is the need for space in the airwaves. All wireless services use spectrum, and more users and higher data rates mean one thing: more bandwidth and spectrum. The problem is that there is only so much spectrum to be had. Once we use it up, there is no more. So, that's the issue today. There is a growing spectrum shortage and its effect will be felt by the wireless providers and the rest of us, as well. What can we do about it?

Spectrum 101

The term "spectrum" refers to the wide range of electromagnetic (EM) space that is taken up by radio waves. It is free space in which all wireless signals travel. Radio waves are a combination of an electric field and a magnetic field that travel together at right angles to one another. Their distinguishing characteristics are frequency and wavelength. Frequency, of course, is the number of EM field cycles per second or Hz. Wavelength is the distance between adjacent wave peaks. Wavelength (λ) is generally expressed in meters and is related to frequency (f) as such:

$$\lambda = 300/f_{MHz}$$

For instance, a 100 MHz FM radio station has a signal wavelength of 300/100 = 3 meters. The higher the frequency, the shorter the wavelength.

The radio spectrum extends 30 kHz to 300 GHz. It's divided into segments as shown in **Table 1**.

As for terminology, any signal over 1 GHz is said to be a microwave signal. Signals in the 30 to 300 GHz range are called millimeter waves. The TeraHertz spectrum above 300 GHz is essentially unused as there are so few actual electronic components that can work at these frequencies. Most transistors run out of gain at about 200 GHz. So, THF is basically a no-man's land.

Beyond 300 GHz is what is known as the optical spectrum. Yes, light is a kind of EM wave. At the lower end is infrared (IR); still used in wireless remote controls and heat sensing applications. Right beyond that is the visible spectrum — red at the low end and violet at the upper end. A little higher in frequency is ultraviolet (uV). Beyond uV are the far-out cosmic and gamma rays.

The most widely used (thereby, the most crowded) spectrum segments are the VHF and UHF bands from 30 MHz to 3 GHz. These areas of the spectrum contain all the mobile radio, cellular, and short-range services like Wi-Fi and Bluetooth. The spectrum between about 300 MHz and 6 GHz is generally known as the "sweet spot." This prime spectrum is popular as it offers a great balance between range or distance of transmission and reasonable (meaning short) antenna length.

It is important to know and keep in mind that radio waves obey the rules of physics that say that the higher the frequency, the

Post comments on this article and find any associated files and/or downloads at **www.nutsvolts.com/magazine/article/December2016_OpenCommunication**.

more limited the range for a given transmitter power and antenna gains. A 2.4 GHz Wi-Fi signal will travel farther than a 5 GHz Wi-Fi signal with all conditions being the same. This is both good news and bad news. The good news is that at some distance, the signal is just too weak to interfere with someone else on the same frequency. The bad news is that the communications range is just too short for reliable service. This feature or limitation allows frequencies to be shared or reused, which is one of the key solutions to utilizing spectrum.

Spectrum Management

Spectrum is sacred ground, so must be protected and used as necessary; that is why it is government regulated. All countries in the world have spectrum regulatory services. In the United States, these are the Federal Communications Commission (FCC) and National Telecommunications and Information Administration (NTIA). The FCC addresses the consumer and commercial spectrum, while the NTIA oversees the military and government spectrum.

The NTIA offers a master spectrum chart (small version shown in **Figure 1**; full-size version available at the article link) that you can download from **www.ntia.doc.gov/page/2011/united-states-frequency-allocation-chart**. This gives you a look at the massive complexity involved in allocating and managing spectrum.

These sites are also worth a visit if you are hungry for more spectrum related material: **www.fcc.gov** and **www.ntia.gov**.

If you want to know more about the spectrum, get a copy of the Code of Federal Regulations (CFR) Title 47 Parts 0 through 100 that contain all of the FCC's rules and regulations. Part 2 covers the frequency allocations for the various services. Parts 15 and 18 cover short-range wireless and electromagnetic interference (EMI) regulations. Go to **https://www.fcc.gov/general/rules-regulations-title-47** for more details.

Keep in mind that there are two types of spectrum: unlicensed and licensed. Unlicensed spectrum can be used without prior FCC approval if the related rules and regulations are precisely followed. Licensed spectrum is strictly awarded and policed by the FCC for specific services such as radio and TV broadcasting, land mobile, cellular, satellite, and radar.

Minimizing the Spectrum Problem

The spectrum limitations have been known for years. Fortunately, the industry has been at work trying to resolve the problem with technological and management solutions. Here are just a few approaches you may not be aware of.

Take advantage of spatial diversity. Spatial diversity is essentially the same as frequency reuse. This is the idea that different services may use the same frequency bands if they purposely avoid others that could interfere. Keeping a distance from one another is the main idea. Use less power to limit range of coverage. Use highly directive gain antennas to focus the communications and reduce interference to others. This also boosts effective radiated power extending the range.

The cellular industry uses spatial diversity to get multiple uses out of its frequency assignments by using sectorized antennas and by controlling handset power. Proper cell site spacing and antenna focusing greatly improve spectrum utilization efficiency.

Move to the higher frequencies. If you need more bandwidth to achieve the desirable higher data rates, go where the bandwidth is: at the higher frequencies. Move from UHF to microwave or from microwave to the millimeter wave bands. There has been a steady shift to the higher frequencies over the years as semiconductor technology has enabled higher frequency devices.

Cellular service has moved from UHF to low microwaves as LTE service expanded. Now, work on the 5G cellular system is producing another move to the millimeter

Name	Frequency Range	Applications
Low Frequency (LF)	30 to 300 kHz	Navigation, time standards.
Medium Frequency (MF)	300 kHz to 3 MHz	Marine/aircraft navigation, AM broadcast.
High Frequency (HF)	3 to 30 MHz	AM broadcasting, mobile radio, amateur radio, short-wave broadcasting.
Very High Frequency (VHF)	30 to 300 MHz	Land mobile, FM/TV broadcast, amateur radio.
Ultra High Frequency (UHF)	300 MHz to 3 GHz	Cellular phones, mobile radio, wireless LAN, PAN.
Super High Frequency (SHF) Millimeter wave range	3 to 30 GHz	Satellite, radar, backhaul, TV, WLAN, 5G cellular.
Extremely High Frequency (EHF)	30 to 300 GHz	Satellite, radar, backhaul, experimental, 5G cellular.
TeraHertz, Tremendously High Frequency (THF) or Far Infrared (FIR)	300 GHz to IR	R & D, Experimental.

bands of 27 to 40 GHz. Wi-Fi is adding 5 GHz and 60 GHz to get the bandwidth for higher speeds. That trend continues as the semiconductor developments make it practical.

Improve spectral efficiency. Spectral efficiency is the concept of squeezing higher data rates out of the same or less amount of bandwidth. A primary way to do this is to use higher order or multicarrier modulation schemes. QAM is a good example as it transmits many more bits per Hz of bandwidth. OFDM as used in LTE cellular and Wi-Fi squeezes much higher rates into a narrow spectrum.

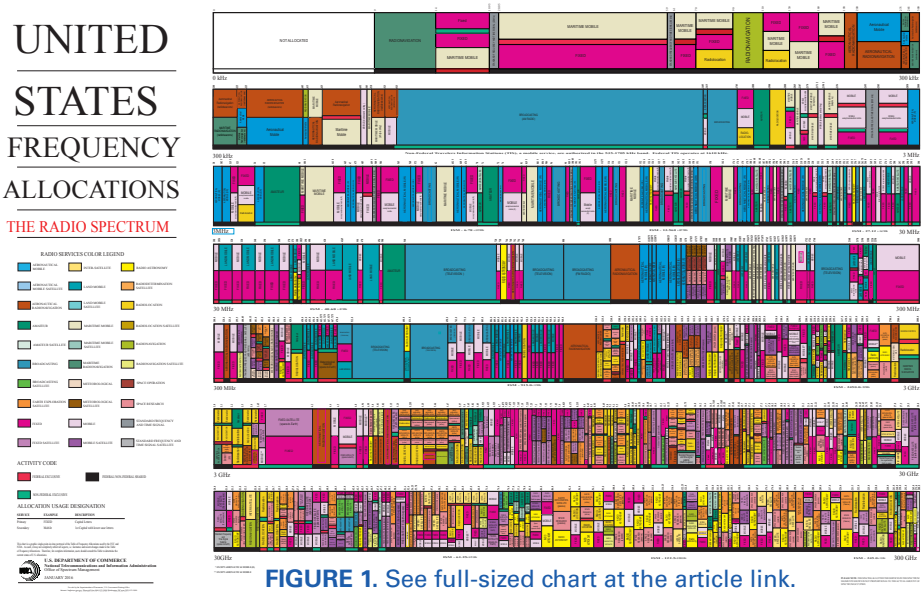


FIGURE 1. See full-sized chart at the article link.

Another technique is data compression that reduces the number of bits to transmit voice, music, or video. A newer method is multiple input multiple output (MIMO) as used in LTE cellular and Wi-Fi. MIMO takes the high speed data and divides it into multiple parallel streams that are then sent to multiple transmitters and antennas. The multiple signals are transmitted in the same frequency band. The differences in transmission paths allow the multiple receivers to recover the data. MIMO not only boosts data speeds, but also improves link reliability.

Allow multiple users to share spectrum. We do this now, and there is more to come. Most of it takes place in the unlicensed spectrum. The 2.4 GHz band is a great example. It is used by Wi-Fi, Bluetooth, ZigBee, and multiple other wireless services such as drone control and video, and even microwave ovens. Problems are mitigated by the short ranges usually involved, but also by way of unique modulation, avoidance, and channel assignment techniques.

Another application is cellular offload. This is a technique where the cellular operators transfer high data rate connections like video to a nearby Wi-Fi hot spot if one is available. This keeps the cellular network from going into overload as it sometimes does. The Wi-Fi 5 GHz band is popular as now the Department of Transportation (DOT) wants to use a portion of that band for vehicle-to-vehicle (V2V) communications. V2V is a forthcoming system that lets cars talk to one another about speed, location, and direction for safety purposes.

Will interference be a problem? A consortium of cellular interests wants to use the 3.5 GHz band for expanding cellular services. Will it conflict with the radar in that band?

Then, there is white space. White space refers to the unused TV channels around the US. Many of these 6 MHz wide channels in the 54 to 790 MHz range were abandoned a while back as stations moved to higher frequency bands with the switch to digital, or ended TV broadcasting. These bands are useful for broadband data services in rural areas, industrial monitoring and control, and other wireless data needs. The problem is how to avoid interference to TV stations on some or adjacent channels.

The answer is cognitive radio techniques that listen before transmitting, then waiting until the channel is free. Cognitive radio also usually employs a database that lists all stations and other services that a transmitter can access on-the-fly to determine the channel's availability.

Spectrum repurposing. Patches of spectrum may not be widely used, but are allocated such that it cannot be used for other applications. Some government and military spectrum is like that. The idea is to get the spectrum owner to give it up for other uses. No one wants to lose spectrum holdings, but some government and military spectrum could be repurposed if they can be convinced it is beneficial.

The FCC is encouraging this approach. It is now conducting auctions for spectrum. TV stations are motivated to give up unused channels as they will receive a portion of the auction income. Most of the buyers pay millions — even billions — to get more bands for cellular expansion.

Final Transmission

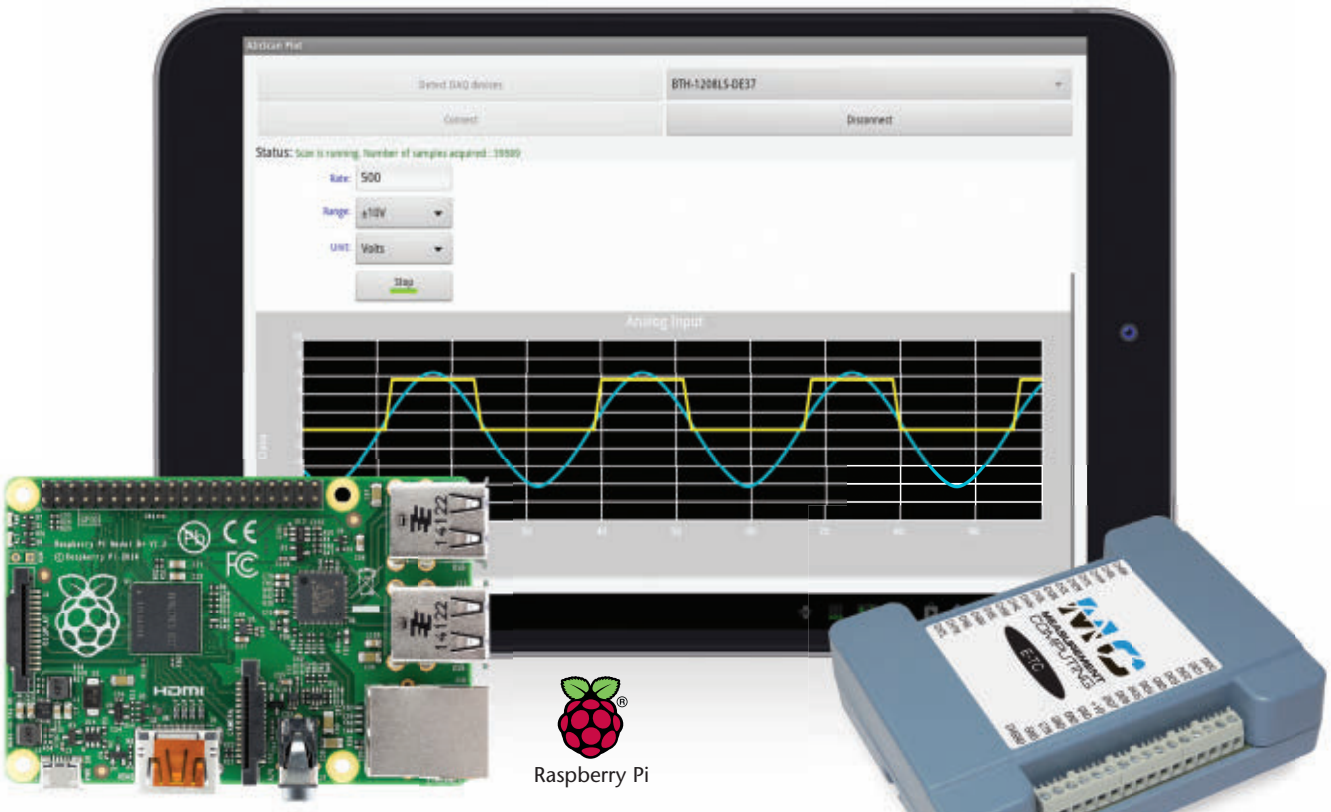
The spectrum shortage is being dealt with one approach at a time. It will only get worse as more wireless applications are being implemented. New technologies and improved spectrum management will continue to resolve problems as they show up. We always find a way to solve them.

Innovation will play a major role as spectrum availability comes to an end. Not right now, but in the future. **NV**

MEASUREMENT COMPUTING

Create Embedded and Portable Data Logging Apps

Easy to Use • Easy to Integrate • Easy to Support



Raspberry Pi

- Android™ drivers for phones and tablets
- Linux® open-source drivers
- Support for over 40 USB, Ethernet, and Bluetooth® DAQ devices

The E-TC offers high-accuracy temperature measurement with an Ethernet interface.

Test drive MCC products risk free with our 30-day return policy. Call for details.

mccdaq.com/Android

MEASUREMENT COMPUTING™

Contact us
1.800.234.4232

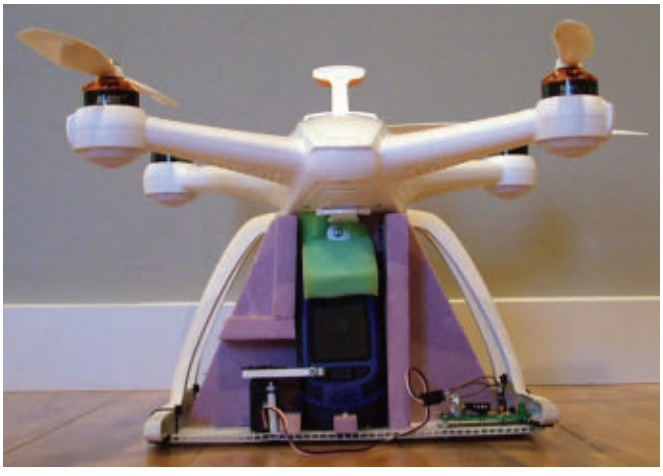
©2016 Measurement Computing Corporation, 10 Commerce Way, Norton, MA 02766 • info@mccdaq.com

■ BY L. PAUL VERHAGE
paul@nearsys.com

Cheap Multispectral Imaging for Amateur Science

Testing imaging methods in preparation for a future near space mission makes me realize just how useful quadcopters and BalloonSats can be in imaging Earth's surface. Satellites and airplanes are the primary platforms in use today. However, satellites are very high above the ground and expensive, while airplanes are just mildly expensive and unable to safely view the ground up close. This means drones like quadcopters are the solution to Earth imaging very close to the ground and BalloonSats are the cheap alternative to satellite-based imaging.

Satellites orbit Earth at altitudes above 250 miles; that is, if they want to remain in orbit for any length of time (although I've read that spy satellites will lower their orbits for higher resolution images). Many airplanes fly upwards of 30,000 feet and as low as 500 feet above the ground in uncongested locations (there are exceptions for airplanes like crop dusters). However, for images taken at altitude below 400 feet and above 60,000 feet, there are no practical and inexpensive platforms except for the quadcopter (drone) and the BalloonSat.



I built the carrier for this thermal imaging system using Coroplast and Styrofoam. The quadcopter then carried it up to an altitude of 400 feet AGL for an aerial imaging experiment. Next stop, near space!

Before discussing my experiences with using quadcopters and BalloonSats to image, I'd like to describe my experiments using instruments to record images in a variety of spectral ranges. I'm testing imaging techniques on the ground and then moving them up to my quadcopter for further testing. The experience gained will then be used to create a BalloonSat to collect similar images from near space.

The types of imagers I've tested so far include a color camera (visible), a near infrared camera, and thermal imager (long-wave infrared). The imager I haven't successfully created is the near ultraviolet camera (but I'll keep working on it).

The Electromagnetic Spectrum

During Isaac Newton's lifetime, physicists debated whether light was best described as a particle or a wave. Newton claimed that the results of his experiment with prisms were proof that light consisted of particles or corpuscles as he described them. Less than a decade later, Christiaan Huygens developed a theory of light that proved light could indeed be a wave and still behave as physicists had observed.

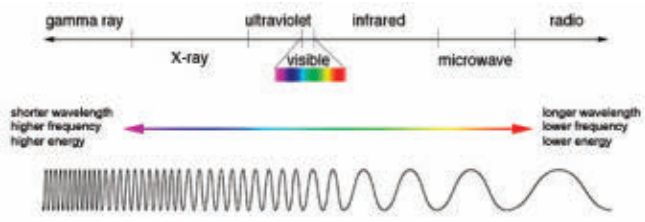
Astronomer William Herschel later extended the range of colors by discovering a new region of the spectrum that the human eye could not see; a color he named infrared (below red). One year later, Johann Ritter did something similar to Herschel and discovered the existence of light above violet. He called this color Chemical light, but it was renamed ultraviolet since it was above violet.

Later in the 19th century, physicist James Clerk Maxell noticed that he could combine two equations that described magnetic fields and two equations that described electric fields into the wave equation. The equation indicated that the combined electric and magnetic wave was self-supporting and traveled at the speed of light. This was strong evidence that light is indeed an electromagnetic wave and that other "colors" of light that we can't see existed. Today,

Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/magazine/article/December2016_NearSpace_Multispectral-Imaging.

astronomers used the entire electromagnetic spectrum to study the cosmos.

Portions of the Electromagnetic Spectrum that Amateurs Can Use



Visible light is only a tiny portion of the entire electromagnetic spectrum. Science has discovered many of the ways that the universe works using tools to measure the electromagnetic spectrum at various wavelengths.

Forming images requires optics. For amateurs, the most practical optics is the lens and mirror. When it comes to portable cameras for drones and BalloonSats with their limited payload weight, lens trump mirrors. Using lenses means amateurs are limited to imaging where lenses are transparent, or between the infrared and ultraviolet portion of the electromagnetic spectrum.

A digital camera makes a convenient imaging system since its lightweight, uses no film, and produces images easily manipulated with software. The CMOS imager in a modern digital camera can detect light in the visible, infrared, and ultraviolet portions of the spectrum where most lenses are transparent. That means a digital camera is capable of imaging part of the near infrared (NIR) between 850 nanometers (nm) and 750 nm, the visible spectrum between 750 nm and 400 nm, and the near ultraviolet (NUV) between 400 and 250 nm.

Near infrared (NIR) imaging — which spans the range of wavelengths between 750 nm and 1,400 nm — is useful for detecting the chlorophyll of plants. Chlorophyll is green because it reflects green photons, and absorbs red and blue photons. While we can't see it, chlorophyll also reflects near infrared. By comparing the amount of red and near infrared light reflected by a plant, we can assess the presence and health of plants.

Satellite images make a mathematical comparison between the intensity of red and NIR called the Normalized Difference Vegetation Index (NDVI). NDVI has a value between -1 to +1. NDVI values of -1 indicate water; values near 0 indicate barren land; and +1 indicates thick foliage. You can read more about this interesting technique at Open Lab's website at **http://infragram.org**.

An easy way to convert a digital camera to NIR imaging is to remove the IR blocking filter inside the camera and replace it with several sheets of red and blue theater gels. Two blue gels will block all visible light except for a small amount of blue. A single red gel will block the remaining blue light, and thereby stop all visible light from reaching the CMOS imager.

Theater gels are designed to transmit infrared light because of the high temperature of theater lighting. Indeed, if the theater gels also blocked infrared, they would get hot enough to melt or even catch fire (kind of a bad thing to happen). So,

A comparison between a tree viewed in near infrared (L) and visible light (R). The tree's leaves are so bright in NIR that it appears the image on the left was recorded on a snowy winter day rather than in the middle of summer. Also, notice how dark the sky is in NIR. That's because molecules of oxygen and nitrogen are approximately the same size as the wavelength of blue light. Their close match in size to the wavelength of blue makes these atmospheric molecules very effective at scattering the Sun's blue light, but note its much longer red and infrared light. Therefore, when an NIR camera is not aimed at the Sun, it sees no scattered sunlight. Since the air does not emit NIR, it leaves the sky very dark.

when theater gels block all of the visible light, only infrared is allowed to pass through.

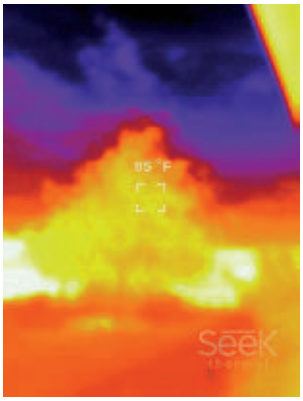
Recently, the long-wave infrared portion of the spectrum has opened to amateur use. Thermal imagers — like the Reveal by Seek — creates and records thermal images in the 8 to 15 micrometer (8,000-15,000 nm) range by using a chalcogenide lens and a micro-bolometer in place of a glass oxide lens and CMOS imager.

For curious readers, the chalcogens are elements in the oxygen family of elements (group 6A elements in the periodic table of the elements). In a chalcogenide lens, oxygen (oxide) is switched with sulfur, selenium, or tellurium. The replacement of the oxide lens of a tradition camera with a chalcogenide one makes the lens transparent to long-wave infrared, which normal glass is not.

The second step in creating a thermal imager is to replace the CMOS imager with a bolometer. A bolometer consists of an array of small cells like a CMOS imager. However, the cells in a bolometer measure the temperature of the cell instead of the intensity of the light shining on them.

In the case of the Seek Reveal, the micro-bolometer is an array of 206 by 156 cells or pixels, so nowhere near the resolution of a modern CMOS imager. Images in the Reveal are coded yellow and white for warm pixels and a blue to black for colder pixels. The resulting image is a temperature map of the object(s) observed.

So far, my experiment photographing in near ultraviolet (NUV) has been a failure. I thought I could just remove the IR (and UV) blocking filter from a digital camera and then slap a UV filter over the camera lens. The UV filter was the glass of a broken black light lamp. The problem is that a black light creates ultraviolet radiation by operating at high temperatures. The bulb's high temperature means the glass must also be transparent in the infrared or else it damages or breaks the glass. The result is that the ultraviolet filter I used transmits both NIR and UV. Therefore, at this time, I'm still looking for a filter that blocks NIR and is visible while transmitting UV.



The tree, now viewed in long-wave infrared. This image indicates that the tree's leaves are warmer than the lawn below the tree. Also, the lawn and road behind the tree are warmer than the tree. A few scattered cool clouds are visible in the cold (therefore, black in color) sky.

There are three cameras and three regions of the electromagnetic spectrum covered so far. I'm interested to hear if any readers are experimenting with amateur multispectral imaging and what tools they're using. In my next article, I'll show some of the results achieved by my quadcopter so far. It's not using a gimbal to support the imaging systems, so the pictures can be blurry at times.

Drone Laws

Drones like quadcopters are a lot of fun to fly. I'm discovering some of the new ways people and organizations are using them to do real science. Unfortunately, some people are expressing a fear for their misuse. I have to admit that quadcopters do make it easier to spy on neighbors or to collect undesired data about farm and ranch operations. In response to these real and perceived threats, some states are passing laws restricting drone use. Earlier this year, I had the opportunity to learn about Idaho's drone law.

While performing the imaging experiments with the tools I'm describing in this article, my quadcopter flashed all four of its status LEDs white — an indication of a GPS error. My quadcopter has a software geofence limiting its flight range to 600 feet. Unfortunately, firmware onboard the quadcopter didn't respond well to the GPS error; it flew away and was non-responsive to my commands.

My quadcopter is registered with the FAA and has my phone number written prominently on it. (That's because I'm a responsible drone pilot and want my expensive quadcopter back should it crash.) Because of its marking, I was confident I'd get a phone call about it during harvest time when a farmer came across it. To my surprise, I actually got the phone call from the Canyon County Sheriff's Office only an hour after the fly-away.

Nope. Can't see inside this house with a thermal imager.

It turns out my quadcopter flew 4,000 feet away (or six times beyond its geofence), and landed in a yard of a concerned citizen who just happened to be my local legislator. It took about seven weeks to get this resolved, and I want to thank the Sheriff's office for handling this. They did a very professional job. Now, all parties understand the landing was unintentional, the result of a failure, and not an attempt to spy.

One concern I've heard voiced because of this incident is that some people believe that thermal imagers can see through walls and into homes. To show this is not the case, I want to show the image here that I took while looking at a window of my house. The curtain was pulled back and I could see inside my house. However, you'll notice that the thermal imager only sees my reflection.

Window glass is not transparent to heat and therefore not transparent to long-wave infrared. However, it is reflective. The only thing a commercial thermal imager can show is the warmth of a surface and not the heat of objects on the other side of a wall or window.

Please check your state laws before flying your drone. I know Idaho and Texas have drone laws that can affect how and where you can fly.

Overall though, I've been very happy with the results. I imagine that quadcopters will soon be using simple imaging techniques like I'm experimenting with to determine the health of crops — especially now that the FAA has created the remote pilot license with small unmanned aircraft systems rating.

Success of my quadcopter will lead me to testing BalloonSats as an amateur multispectral imaging system and I'll share those results later as well.

Onwards and Upwards,
Your near space guide **NV**

ALL ELECTRONICS
www.allelectronics.com Order Toll Free 1-800-826-5432

12VDC SPDT 40A AUTOMOTIVE RELAY
12 Vdc, 88 ohm coil. Contacts rated 40A. Plastic enclosed relay, 1.1" x 1.2" x 1" high. Plastic flange for bulkhead mounting. Mounts in standard automotive relay socket.
CAT# RLY-351
$2.55 each
10 for $2.25 each

1W LED W/ REFLECTOR IN HOUSING
Bright, white 1W LED enclosed in a rugged black and yellow hard plastic housing. 2.16" dia. x 3.45" long. Operates on 3 Vdc @ 300 mA.
CAT# LMP-26 $2.50 each
10 for $2.25 ea.

3-PRONG GROUNDING AC PLUG
Leviton # 515PR.
15 Amp grounding 3-wire AC plug. Rubber body is dirt & moisture resistant. Tough elastomeric husk. Built-in cord grip adjusts to 0.245" to 0.655" cords, 18-3 to 12-3. UL, CSA.
CAT# ACP-8
$2.75 each
10 for $2.50 each

12V HIGH-BRIGHTNESS LEDS
5mm (T1-3/4) LEDs designed for 12Vdc operation. No external resistor required. Operate on lower voltages with lower brightness. Ideal indicator lamps, special effects, vehicle or model lighting.
RED w/ red diffused lens CAT# LED-12R
WHITE w/ water-clear lens CAT# LED-12WP
50¢ each
10 for 45¢ each
100 for 40¢ each

HOOK & LOOP STRIPS
4" x 0.90" hook and loop strips with strong adhesive backing. For indoor/outdoor use. Water resistant. Includes both hook and loop. Package of 12.
CAT# HL-412
$2.50 12-pack
10 for $2.25 per 12-pack

18 AWG STRANDED WIRE
35' roll of 18 gauge stranded, black, multi-purpose wire. Suitable for home automotive and electronic project use. UL, CSA.
CAT# 18BK-35
$4.30 per roll

2" BOSTON ACOUSTICS
2" cone with soft rolled edge. 4 Ohm impedance. Shielded magnet.
CAT# SK-304
$2.00 each
10 for $1.75 each

INFRARED SENSOR
RE 200B is a passive infrared sensor designed to pick up heat radiation in the infrared range. Used in alarm systems and automatic switches that respond to human body motion. TO5 metal case w/ 3 leads.
CAT# PIR-200B $1.00 each
10 for 90¢ each

PoLabs
For more information or software download please visit
www.poscope.com

PoKeys
Connect, Control
- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC

Stepper motor drivers
Drive
- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A

PoScope Mega1+
PoScope Mega50
Measure
- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

6WD ROVER
MINI MANTIS

The new Mini Mantis™6WD rover from ServoCity has a bug-like chassis which offers extreme A-arm style suspension and 3.3125″ of independent wheel travel. With 485 RPM Economy gear motors, there is plenty of torque to turn the aggressive 4.3″ off-road tires even when up against the toughest obstacles.

The suspension incorporates 3.85″ aluminum beams and ServoCity's 130 mm oil-filled aluminum bodied shocks. The 11.75″ wide (full extension), 14.75″ long, 6.875″ tall (full extension) boxed channel chassis provides a rigid backbone with endless mounting and customization options, making it easy to bolt Actobotics® components and electronics directly to the chassis of the rover.

Rubber grommets and end caps are included for protecting the motor wires as they get routed to the motor controller of choice. The 1.90" black robot wheels are driven using 12 mm aluminum hex wheel adapters to ensure a solid no-slip connection. Price for the Mini Mantis is $299.99.

V-WHEEL
KIT

The new ball-bearing V-Wheel kit also from ServoCity works in conjunction with 80/20® (1010 series) as well as ServoCity's own X-Rail to create smooth linear motion. The Acetyl construction of the V-Wheels offers high durability and wear resistance. The internal ball bearing has an ID of 5 mm and the overall V-Wheel diameter is 0.60″. Four 5 mm to 1/4″ tapped standoffs are included to make it easy to attach the V-Wheel to channel or other Actobotics components. The 0.675″ standoff length makes it ideal for more compact projects. The kit — which is priced at $29.99 — includes:

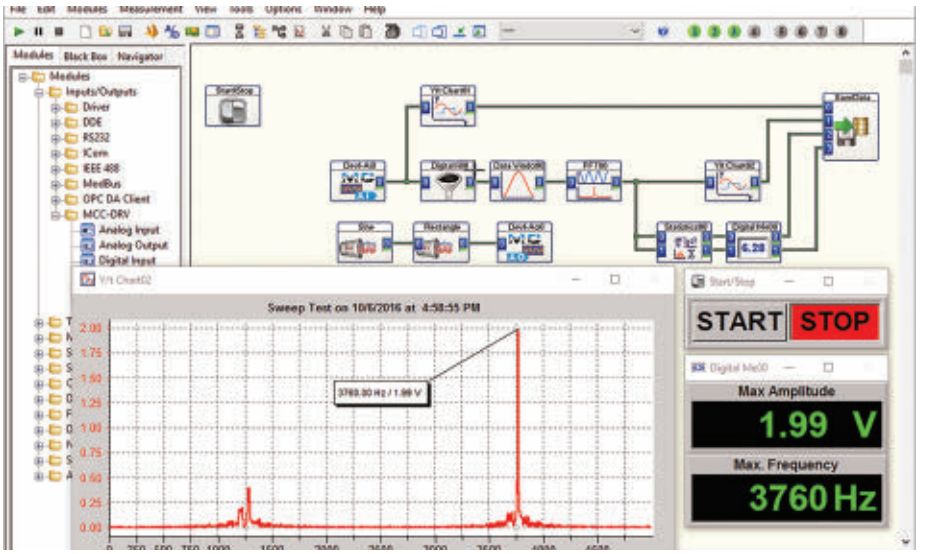
- (4) Ball-bearing V-Wheels
- (4) Standoffs
- (4) #6 Washers
- (8) 6-32 x 1/4″ socket head cap screws,

For more information, contact:
ServoCity
www.servocity.com

OFF-ROAD ROBOT
TIRES WITH WHEELS

New 85 mm wheels/tires from ServoCity are available in blue, red, and silver. These wheels/tires help make it easy to create compact robot platforms. Cost is $6.99/pair. The product weight is 3.95 oz (0.246 lbs). The outside diameter is 85 mm and the width is 38 mm. Hub style is 12 mm hex.

If you have a new product that you would like us to run in our New Products section, please email a short description (300-500 words) and a photo of your product to: **newproducts@nutsvolts.com**.



ICON BASED DATA ACQUISITION SOFTWARE

Measurement Computing Corporation (MCC) has now released DASYLab 2016. DASYLab is a data acquisition software application that allows users to configure custom acquisition, analysis, and control applications in minimal time and without programming. New enhancements include a new state machine module, support for double-precision data, larger block sizes, and more.

DASYLab boasts a very short learning curve. Users interactively develop PC based data acquisition applications by simply attaching functional icons. Real time displays, math and analysis, and control functions are all included.

DASYLab can be used with Measurement Computing, Data Translation, and IOtech brand DAQ devices from MCC, as well as hardware from other suppliers. A free evaluation version of DASYLab is available for download.

New features include:

• State Machine Module. Easily set up sophisticated sequences more easily and efficiently. Users can reorder steps, or insert, delete, or append steps at any position in the state machine.

• Support for Double-Precision Data. Internal data format changed from a 32-bit float to a 64-bit double for data connections between modules. This structure enhancement supports high resolution measurements and significantly increases accuracy.

• Larger Block Sizes. Block sizes up to 1 ms are now supported. Support of larger block sizes increases the resolution and accuracy of block-related operations such as FFT.

• Support for TDM/TDMS File Format . Users can read/write National Instruments (NI) technical data management (TDM) files and TDM streaming (TDMS) files. TDMS files combine the benefits of several data storage formats — ASCII, binary, XML, and database — into one file format.

• Enhanced DASYLab DDF File Format. DASYLab 2016 can read and store double-precision data. DASYLab saves all data channels that contain measurement data, TTL signals, histogram data, and calculation and scaling results in a binary DASYLab data format (.ddf) file.

• Support for Python 2.7.10. The Script module supports Python 2.7.10 and many Python libraries in the public domain.

• Support for Virtual Time Bases. Users can refer data sources such as a switch, slider, or generator module to a virtual time base, and then change the synchronization to the real time base for all modules at once.

• Support for Windows® 10/8/7 32/64-bit.

Current users of DASYLab should contact Measurement Computing about upgrade options for their current version of DASYLab. Pricing begins at $499 with the DASYLab Lite version.

For more information, contact:
Measurement Computing
www.mccdaq.com

The arLCD 3.5" smart TFT touchscreen combines Arduino UNO with the ezLCD smart LCD.

earthlcd.com/nuts

By Ron Hoffman

Build a Sump-Fill Level CONTROLLER

Recently, someone told me that they were having problems with their sump pump. The float had a leak causing it to sink, or else the float switch was bad. He wasn't quite sure what the problem was. At any rate, the sump pump failed to work and resulted in a back-up that flooded their basement and ruined many of their possessions stored there. Another friend said that he wanted to control an elevated cistern that fed plants in a hydroponic garden.

After some thought, I figured that a unit could be designed that could do both jobs. Further, two different approaches could be used. One would depend on the conductivity of water; the second would rely on pressure created by the depth of a liquid in a vessel. Why, you ask, would both strategies be required?

The short answer is that diesel fuel, gasoline, oil, and other types of non-ionic fluids would not work with a circuit that required fluid conductivity. In addition, if the solution were highly caustic or acidic, it would dissolve the sensing electrodes.

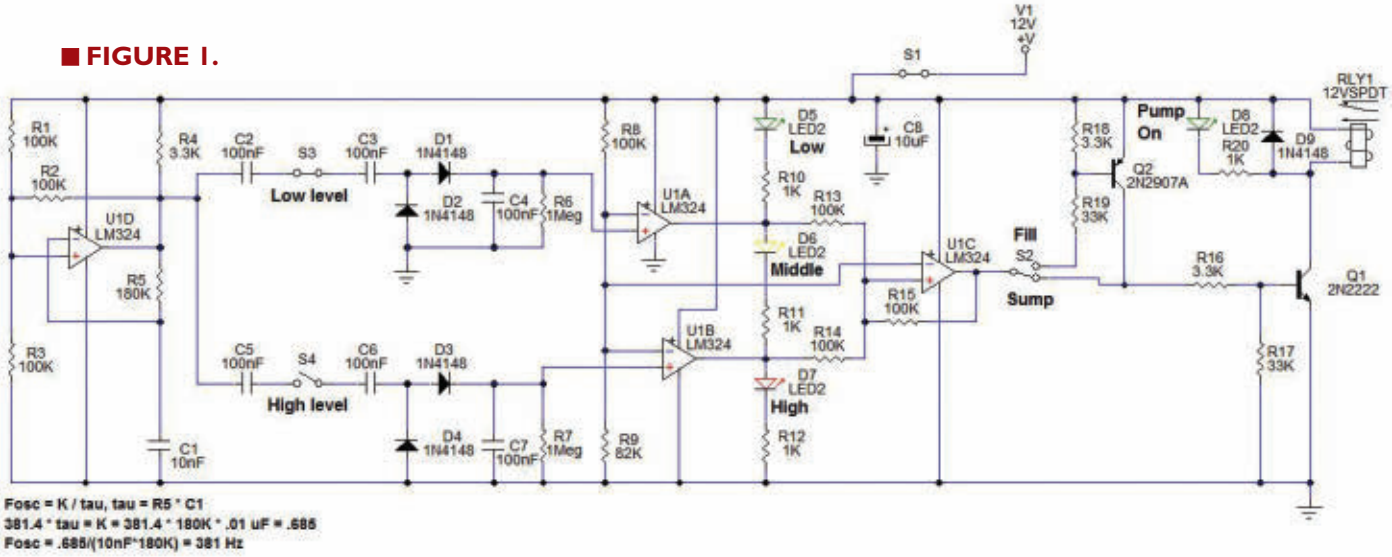
The pressure sensing type of controller can be used with any type of liquid. So, you ask again, why bother with the ionic type of controller? For water and water solutions with dissolved fertilizer, for example, the ionic level controller works fine and costs much less than the pressure level controller. The pressure level controller has the advantage of working with any type of liquid, but is more costly to make.

The great advantage to either controller is that they don't have any moving parts, linkages, floats, or switches that can stick or fail to cause a problem. Also, by negating the output of the logical window comparator, the user can incorporate a simple two-pin jumper to select either the sump or fill mode of operation. One unit can be used for either the sump pump or tank-fill function.

Theory and Operation

Please refer to **Figure 1** (schematic of ionic level controller) for this discussion. The ionic sump-fill level controller uses two electrode pairs — low level and high level (jump ahead and check out **Figure 7 A/B/C** that shows the electrode assembly and control unit) — to detect the liquid level in a vessel such as a sump or storage tank. The electrodes are represented in **Figure 1** as S3 and S4.

■ **FIGURE 1.**



In order to prevent the sensor electrodes from degrading in the ionic water solution due to plating effects, an AC (alternating current) source is used to stimulate the sensor electrodes. Op-amp U1D is configured as a square wave oscillator. Resistor R1, R3 form a Vdd/2 resistive divider for the + input of U1D. Resistor R2 provides positive feedback to the U1D+ input; R4 pulls up the output of U1D to Vdd. Resistor R5 and capacitor C1 form a delay time constant to the - input of U1D.

When the output of U1D is low, the + input of U1D equals Vdd * (R3//R2)/((R3//R2) + R1) = Vdd * 50K/150K = 1/3*Vdd. Resistor R5 discharges C1 until the voltage at U1D ≤ U1D+. The output of U1D switches from low to high. The U1D+ input switches to Vdd * R3/((R1//R2) + R3) = Vdd * 100K/150K = 2/3 * Vdd. Resistor R5 begins charging C1 until the voltage at U1D ≥ U1D+. This process repeats, producing a square wave at the output of U1D. With the values shown, Fosc = approx. (.685 /(R5 * C1)) = 381 Hz (400 Hz was the target).

The output of U1D is fed to capacitors C2 and C5 which are connected to one side of the low and high level electrode sensing pair, respectively. The other side of the low and high level electrodes connects to C3 and C6, respectively. Capacitor C3, D1, D2, C4, and R6 function as a charge pump. When solution allows current to flow from C2 to C3, C4 charges to indicate the solution level is at least as high as the low level electrode pair. Similarly, C5, C6, D3, D4, C7, and R7 detect the high level of the solution in the vessel.

Resistors R8 and R9 form a voltage divider that is half of Vdd - 1.5V (due to the limited output level of the LM324 op-amps). Whenever the voltage level on C4 or C7 is above the R8, R9 reference level, op-amps U1A and U1B (that work as comparators) switch from low to high. When both comparators are high, this overcomes the hysteresis of U1C and causes U1C's output to go high.

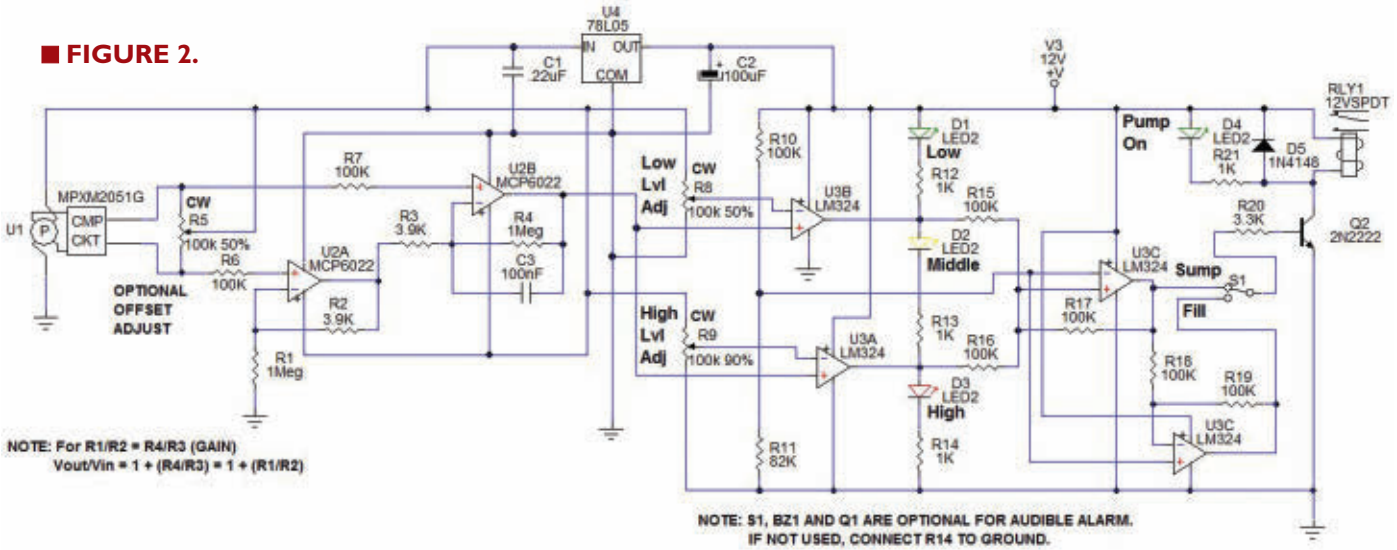
The operation in the sump mode proceeds as follows. The sump level is below the low electrodes; therefore, C4 and C7 are discharged, and U1A and U1B outputs are low. This causes LED D5 to be ON indicating the sump level is low. U1C is low; therefore, RLY1 is OFF (so is the sump pump).

When the solution rises above the low level electrode pair, C4 charges and U1A's output goes high. This turns OFF D5 and turns ON D6 (because U1B's output is still low), indicating the tank level is somewhere between the low level and high level electrodes. U1C stays low due to its own positive feedback and the low state of U1B.

The +input of U1C = (Vdd-1.5V)* R15//R14/(R15//R14+R13) = 10.5 * 50/150 = 3.5V which is below the reference of 5.4V. When the solution rises above the high level electrode pair, C7 charges and U1B's output goes high. This turns OFF D6 and turns ON D7 (because U1A's output is still high), indicating the tank level is above the high level electrode pair. The +input at U1C is now (Vdd-1.5V)* (R15)/(R13//R14+R15) = 10.5 * 100/150 = 7V which is above the reference of 5.4V. This causes U1C's output to go high that, in turn, activates Q1, RLY1, turning ON the sump pump. As the pump lowers the tank level, U1B goes low. This turns OFF D7 and turns ON D6, indicating the mid level.

The + input at U1C = (Vdd-1.5V)* (R14)/(R15//R13+R14) = 10.5 * 100/150 = 7V which is still above the reference of 5.4V. So, that U1C's output remains high, RLY1 stays ON, and the sump pump keeps running. When the tank level is lower than the low level electrode pair, U1A's output goes low, D6 turns OFF, and D5 turns

■ **FIGURE 2.**



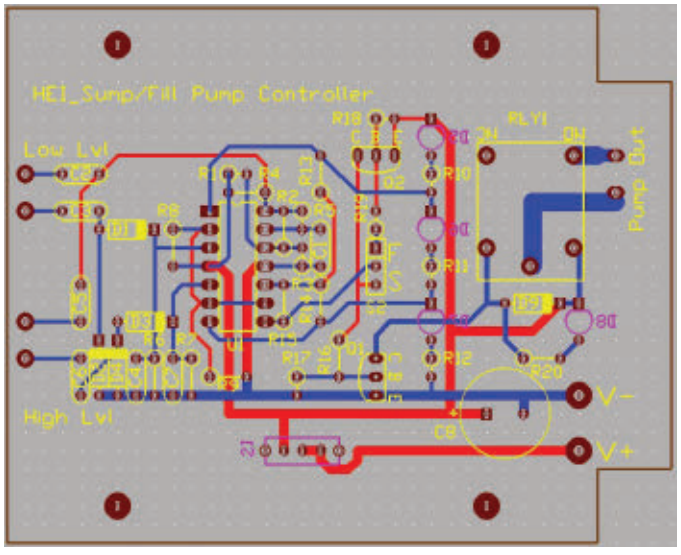
ON (indicating a low level in the tank). When U1A and U1B are low, the + input to U1C = (Vdd-1.5V)*(R13//R14)/(R13//R14+R15) = 10.5*50/150 = 3.5V, causing U1D's output to go low, turning OFF RLY1 and the sump pump.

S2 represents the three-pin jumper selector. The fill mode is selected by connecting the jumper from the output of U1C to R19 and the base of Q2. This negates the output of U1C so that when U1C is high (the tank level is high), Q2 is OFF and Q1, RLY1, and the fill pump or fill valve solenoid are OFF. When the tank is emptied below the low level, U1C goes low, which turns Q2, Q1, RLY1 ON to activate the fill pump or fill valve solenoid to refill the tank. The pressure sensor level controller performs the same sump pump or tank fill function, but does it in a different way. Please refer to **Figure 2** for the following theory and operation discussion.

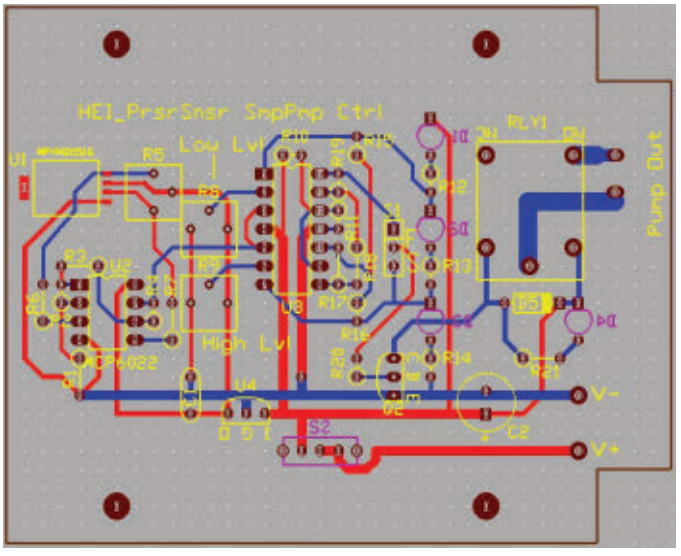
The pressure sensor level controller uses a Freescale MPXM2051G sensor to measure the pressure created by the liquid level in a sense tube that goes from the top to the bottom of the sump or cistern vessel. It works like an inverted drinking glass when you place it into water. The liquid level inside the glass rises as the glass is put lower into the water. The air trapped inside the glass is equal to the water pressure at the bottom (lowest edge) of the inverted glass.

If a flexible tube was connected between the top of the drinking glass (the air bubble inside the glass) and a pressure sensor, the pressure measured would represent the liquid level, or depth of the liquid in the vessel.

■ **FIGURE 3.**



■ **FIGURE 4.**



Instead of a drinking glass, a PVC (plastic), stainless steel, or glass tube is used to trap the air and is connected to the pressure sensor by a flexible tube. (Again, jump ahead and check out **Figure 8** that shows the sense tube, flexible connecting tube, and the controller unit.) The type of sense tube material is determined by the type of fluid that is being measured.

As the pressure rises, the pressure sensor — which is based on a piezo-resistive bridge — puts out a small differential voltage that linearly corresponds to the pressure it senses. The math works out as follows.

The sensor has a full scale range of 50 kPa. Since 1.0 kPa (kilo Pascal) equals 0.145 PSI, the full scale range is 7.25 PSI. The MPXM2051G is ratiometric with the supply voltage. It is rated at 40 mV full scale with 10V excitation.

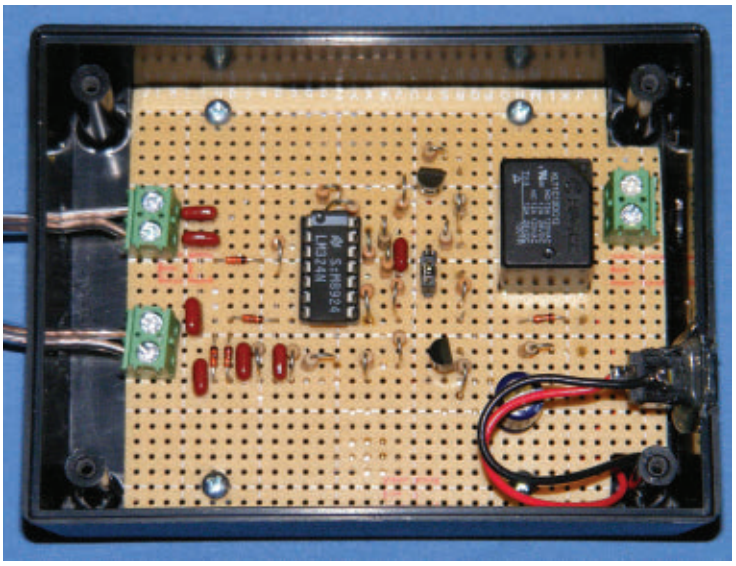
In this design, to accommodate the MCP6022 dual op-amp (used as a zero referenced, differential instrumentation amplifier) five volts of excitation voltage provided by a 78L05, U4 is used. This means that the full scale output of the sensor is 20 mV at 7.25 PSI. Therefore, since 1 PSI = 27.68 in of water, the full scale value of 7.25 PSI = 27.68 in *7.25 = 200.7 in of water for a 20 mV output; 20 mV / 200.7 in H2O = .09965 mV / in H2O. Thus, about 15 feet of water would produce about 18 mV of sensor output.

To get a full scale output from the differential amplifier, a gain of roughly 250 is needed. The gain is (R4/R3 + 1) = (1M/3.9K + 1) = 257.4, so the full scale output is 4.62V. This is fine because the MCP6022 is an RRIO (rail-to-rail, input and output) op-amp. The output of U2B provides a common input level to the U3A and U3B window comparators. Trim pot resistor R8 provides the low level reference to U3A, and trim pot resistor R9 provides the high level reference to U3B.

For a differential level of six inches of water, the differential voltage from U2B is about .09965 * 6 * 257.4 = 154 mV. To prevent noise from causing erratic operation, capacitor C3 in parallel with R4 form a low pass filter with a pole at 6.2832 * .1 µF * 1 Mohm = 1.6 Hz, or a time constant of .1 sec. Resistor R5 is included to adjust the offset of U2B output to around .3V to .5V, to increase the span of level adjustment.

Building the Sump/Fill Controllers

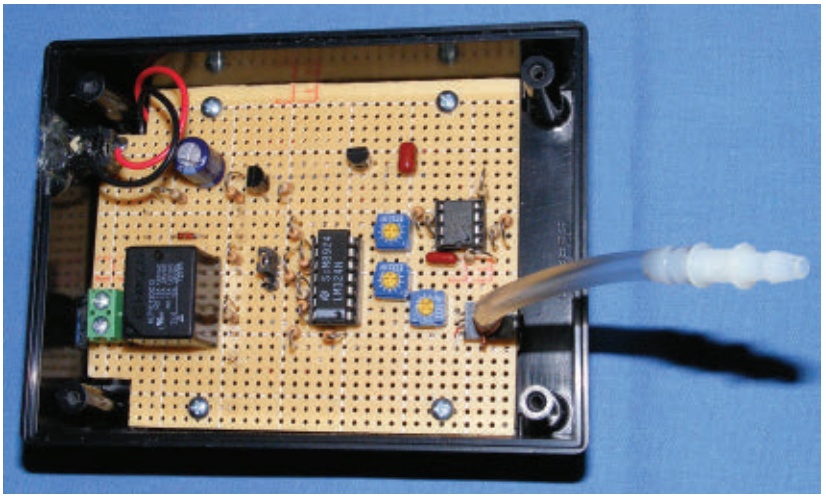
The intention here was to use inexpensive readily-available parts to build the controllers. All of the resistors are carbon film 5%. I chose a SERPAC model 032 plastic case which I have used in previous projects. The board was sized to use the mounting bosses in the case. For the ionic level controller, three wire terminals are mounted on the PCB (printed circuit board) and have slots in the side of the enclosure to allow for the insertion of the sense wires and the pump/solenoid load wires. The power jack (a 2.1 x 5.5 mm barrel jack) was mounted to the side walls of the case. A 12V/1A switching power adapter is used to provide the needed power for the controller.



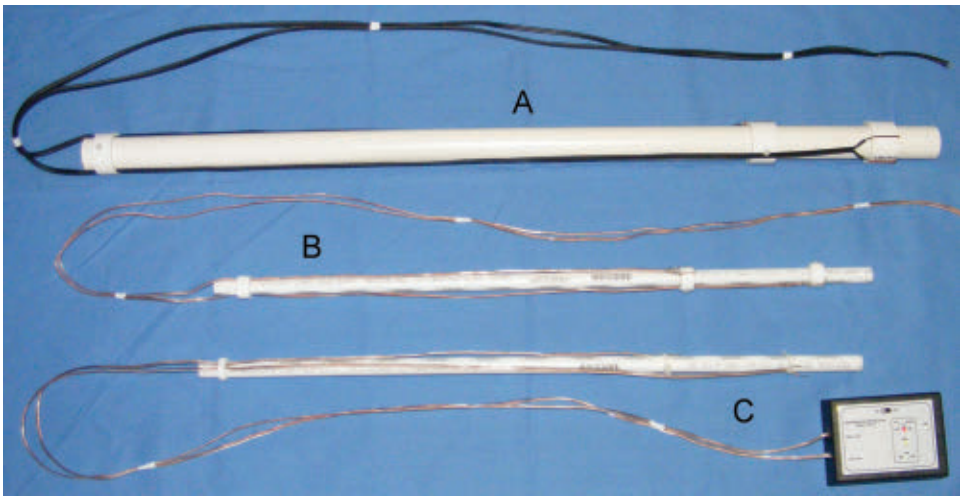
■ **FIGURE 5.**

The Gerber files for a two-sided PCB layout (refer to **Figure 3**, ionic; **Figure 4**, pressure) are available at the article link. For the build, point-to-point wiring was used on a prototype board from RadioShack and cut to the desired size to fit in the enclosure (refer to **Figure 5**, ionic; **Figure 6**, pressure). The components were laid out and positioned the same as on the PCB layout.

After wiring and testing the circuit board, the AC out and input terminals were each connected to their sensor electrodes and load (pump). The power jack was installed



■ **FIGURE 6.**



■ **FIGURE 7.**

into the enclosure, then the wires were connected from the board to the jack. The enclosures were machined using the respective machine labels that are available at the article link.

Measure the center point of each side of the enclosure. The machine labels have center lines that are used to align the labels with the enclosure. The same must be done for the back panel of the pressure unit. The pressure unit has two labels: one for the enclosure front and one for the back. Both the switch and LED holes for the front of the enclosure and the holes for the adjustments and the pressure tube for the back panel are shown on the machine label as an X-ray view. The mirror image label is for the back of the enclosure.

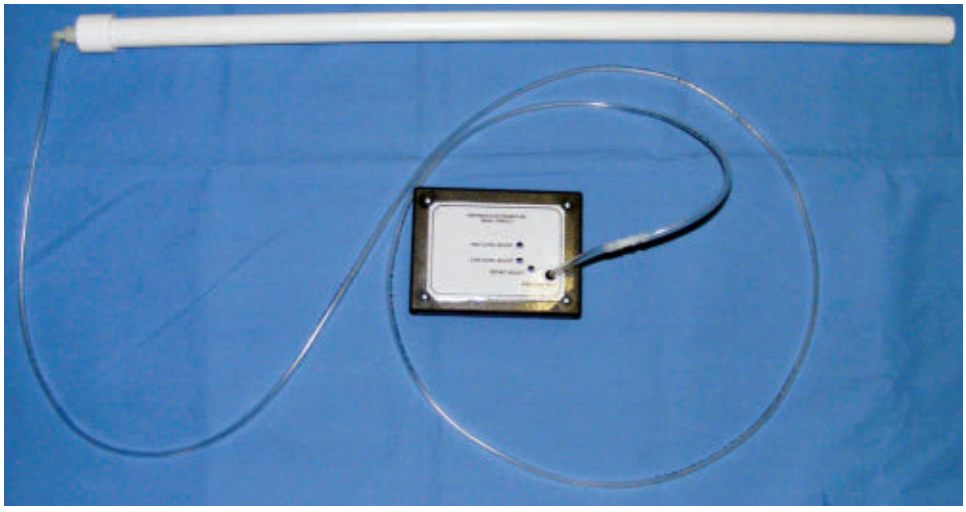
Again, the actual labels used on the enclosure are available for download at the article link. The labels will print out at actual size using 300 dpi in Paint Shop Pro. Since they are jpg files, you can use any photo program using the dimensions of 3.5″ wide by 2.5″ high for the labels, and 3.5″ wide by 2.7″ high for the machine labels.

Several different types of electrode assemblies for the ionic level controllers were built and tested (**Figure 7 A**/**B**/**C**). One electrode probe (**Figure 7A**) was made using one inch and 1.25″ diameter PVC pipe. The 1″ pipe was used as a mounting shaft, while the 1.25″ pipe was cut into two .75″ rings used as the adjustable electrode holders.

7The .75″ rings were drilled and tapped with a 6-32 thread to accommodate a .5″ x 6-32 nylon pan-head screw that is used to clamp the rings to the mounting shaft at the desired liquid level.

Two .0625″ holes were drilled through the wall of the rings to mount a 14 AWG copper wire used as the sense electrode. The 20 AWG SPT-1 lamp wire was soldered to the copper sense electrodes, then run up the mounting shaft through a third ring at the top of the mounting shaft. This kept the wires taut along the shaft. The assembly worked fine, but seemed bulky and difficult to make.

A second version (**Figure 7B**) was built using a .5″ and .75″ diameter PEX tubing. A smaller 24 AWG speaker wire was used as the sensor wire. This assembly was much smaller, less costly, and worked fine too. The rings were machined like the PVC pipe and took some time. A third electrode assembly (**Figure 7C**) was built using a .5″ PEX tube, 24 AWG speaker wire, and three nylon zip ties. This was the easiest to make. The only trade-off was that it is more difficult to adjust the liquid level height. Since the liquid level height adjustment only has to be done during the initial setup of the system, this is not too much of a problem. My preference is either version B or C.



■ **FIGURE 8.**

The pressure based level controller required a sense tube to capture the pressure caused by a rising liquid level (see **Figure 8**). For testing and development, a .5″ PVC pipe was used. The bottom was cut off at an angle to allow the liquid to enter even when the tube was inserted to the bottom of the tank (in this case, a five gallon pail). A .5″ PVC pipe cap was drilled and tapped for a .25″ NPT pipe thread. A .25″ NPT to .125″ barbed elbow was threaded into the PVC pipe cap using Teflon tape to ensure an air-tight fit.

A .125″ flexible tube (i.e., Tygon, or silicone, etc.) was used to connect the elbow to the controller unit. A short 3″ length of tubing was attached to the pressure sensor using a 20 AWG solid copper wire wrapped twice around the tube at the pressure sensor, then twisted with pliers to cinch it tight. A dual .125″ barbed fitting was used to connect the short tube to the long flexible tube coming from the sense tube elbow.

Testing the Liquid Level Controllers

To test the level controllers, the sump fill jumper was connected in the sump position on both controllers. A five gallon pail was used as the test vessel and filled nearly to the top with water (about 16″). A 12 VDC at 1A switching power adapter provided the necessary voltage to operate the level controllers. The ionic controller sense tube (shown in **Figure 7C**) was lowered into the tank to simulate the water rising in the sump tank. When the low level set of electrodes touched the water, the low level LED turned OFF and the mid level LED turned ON; the pump remained OFF.

Continuing to lower the sense tube into the water, the high level electrodes touched the water, the mid level LED turned OFF, the high level LED turned ON, the pump run relay and pump run LED turned ON. The sense tube was raised to simulate the pump lowering the water level in the sump tank. As the high level electrodes rose out of the water, the high level LED turned OFF, the mid level LED turned ON, and the pump relay and LED remained ON.

As the low level electrodes emerged from the water, the mid level LED turned OFF, the low level LED turned ON, and the pump run relay and pump run LED turned OFF. By adjusting the position of the sense electrodes, any low level and high level could be set. I placed the sump/fill jumper in the fill position. The unit worked to display the levels as before, but the pump run relay and pump run LED worked in the opposite way, i.e., they turned OFF when the high level was sensed and turned ON when the low level was sensed.

Next, the pressure based level controller was tested. The sense tube was inserted four inches into the water and held at that position. The low level pot was adjusted until the mid level LED came ON, then backed off slightly until the low level LED turned ON and the mid level LED turned OFF. The tube was then inserted 10 inches into the water and held at that level.

IONIC SUMP

#	QTY	PART REF	DESCRIPTION
1	7	R1, R2, R3, R8, R13, R14, R15	100K resistor 5%, YAGEO, DK 100KQBK-ND or equiv.
2	3	R4, R16, R18	3.3K resistor 5%, YAGEO, DK 3.3KQBK-ND or equiv.
3	1	R5	180K resistor 5%, YAGEO, DK 180KQBK-ND, or equiv.

PRESSURE SUMP

#	QTY	PART REF	DESCRIPTION
1	2	R1, R4	1 meg resistor 5%, YAGEO, DK 1.0MQBK-ND or equiv.
2	3	R2, R3	3.9K resistor 5%, YAGEO, DK 3.9KQBK-ND or equiv.
3	1	R5	100K trim pot, or equiv., DK CT6EP104-ND

PARTS LIST for Both

#	QTY	PART REF	DESCRIPTION
4	2	R6, R7	1 meg resistor 5%, YAGEO, DK 1.0MQBK-ND or equiv.
5	1	R9	82K resistor 5%, YAGEO, DK 82KQBK-ND or equiv.
6	4	R10, R11, R12, R20	1.0K resistor 5%, YAGEO, DK 1.0KQBK-ND or equiv.
7	2	R17, R19	33K resistor 5%, YAGEO, DK 33KQBK-ND or equiv.
8	1	C1	.01 µF metalized poly, ± 10%, Panasonic ECQ-V1H103JL, DK P4513-ND or equiv.
9	6	C2, C3, C4, C5, C6, C7	.1 µF metalized poly, ± 10%, Panasonic ECQ-V1H104JL, DK P4525-ND or equiv.
10	1	C8	330 µF alum elec, 16V, Panasonic, ECA-1CM331, DK P5140-ND or equiv.
11	5	D1, D2, D3, D4, D9	1N4148 silicon diode or equiv., DK 1N4148FS-ND
12	2	D5, D8	LED, GRN (green), Lite-On LTL-4232N or equiv., DK 160-1083-ND
13	1	D6	LED, YEL (yellow), T-1, Lite-On LTL-4252N or equiv., DK 160-1082-ND
14	1	D7	LED, RED (red), Lite-On LTL4222N or equiv., DK 160-1081-ND
14	1	JK	5.1 x 2.1 mm barrel jack, DK CP-037A-ND
15	1	Q1	PN2222A Fairchild Semi or equiv., DK PN2222AFS-ND
16	1	Q2	PN2907A or equiv., DK PN2907ATFCT-ND
17	1	RLY1	Relay, Hasco KLT1C12DC12, SPDT, 12A, 12 VDC, 300 mW or equiv.
18	1	S1	SPDT slide switch, TE Connectivity, DK 450-1609-ND
19	1	S2	Three-pin header, Sullens DK S1212-03-ND
20	1	S2	Two-pin shunt jumper, DK 3M9580-ND
21	1	U1	LM324N quad op-amp or equiv., DK 296-1391-5-ND
22	1	PCB	Printed circuit board, FR-4, two-sided, 2 oz copper (final), per Gerber files
23	1		Enclosure, SERPAC 032-B, Mouser 635-032-B

The high level pot was adjusted just until the mid level LED turned OFF, the high level LED turned ON, and the pump run relay and pump run LED turned ON. As the sense tube was withdrawn from the pail, the mid level LED turned ON and the pump relay and LED remained ON. As the sense tube was removed less than the four inch set point, the mid level LED turned OFF, the low level LED turned ON, and the pump run relay and pump run LED turned OFF.

Applications and Using the Liquid Level Controllers

Obviously, replacing the float and switch assembly in a sump pump tank is one application for the level controllers. Another would be for pumping out water from a dry well. A public water cistern which is mounted above ground to provide local water pressure could be refilled from a well whenever the tank level gets too low.

A top-down hydroponic feeding tank could be refilled to maintain a level between two specified points. The lower hydroponic catch tank could automatically refill to maintain a level between two specified points. Using a unit for the top and bottom tanks would allow for nearly complete automated hydroponic gardening. The only manual task would be adding fertilizer to the system as needed.

Another suggestion would be for maintaining the level of water in an aquarium. By plumbing a small fill line like those used for a refrigerator ice maker, the ionic level controller could energize a small solenoid valve to maintain the aquarium water at an optimum level for the filter pump and aeration system.

The pressure sensing level controller could be employed to fill an above ground fuel tank from an underground storage tank. The controller could be used to auto fill tanker trucks carrying fruit juices or vegetable oils. As was mentioned at the beginning of this article, any type of liquid can be level controlled using these two units to either fill or empty any size container.

The best part is that there are no mechanical floats or switches to deal with, and the level sensing is done remotely which keeps the electronics away from the fluids that are being monitored. As a reminder, the Gerber files and label files for this project are available at the article link. I hope you find this unit useful for many different applications. NV

Technologic Systems

yocto

QNX

TS-7970

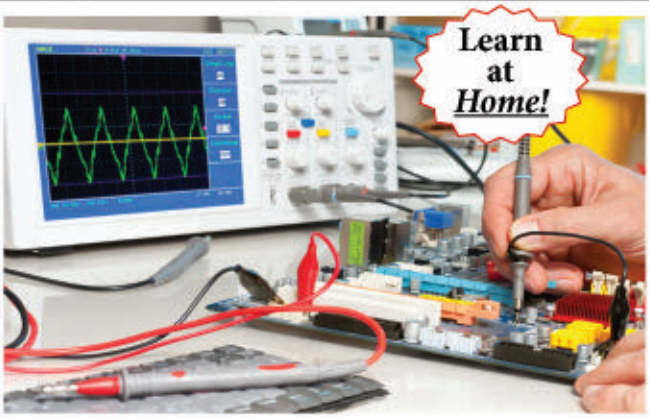
Industrial High Performance i.MX6 Computer with Wireless Connectivity and Dual GbEth

$169 QTY 100

- 1 GHz Single or Quad Core Cortex A9 ARM CPU
- 2 GB DDR3 RAM (Quad) and 4 GB MLC eMMC Flash
- Bluetooth 4.0+EDR and WiFi 802.11BGN onboard radios

www.embeddedARM.com

LONG LIFE OPEN RUGGED ORIGINAL



Electronics Courses

Cleveland Institute of Electronics

Train at home to be a professional *electronics* or *computer* technician! Learn with hands-on labs, instructor support, online exams, videos and instructor led chat rooms.

FREE course catalog www.cie-wc.edu or (800) 243-6446

- NEW! Robotics Automation Lab
- Industrial Electronics with PLC
- Electronics with FCC
- Electronics Troubleshooting
- NEW! Computer Security Specialist
- Broadcast Engineering
- PC Troubleshooting
- Networking

Visit www.cie-wc.edu and start today!

CIE: 1776 E. 17th St., Cleveland, OH 44114 • (800) 243-6446 • Registration 70-11-0002H

Take a CAN Bus for a Spin

Learn how to use a CAN bus with a Propeller MCU

Part 1

By Jon Titus KZ1G

Post comments on this article and find any associated files and/or downloads at **www.nutsvolts.com/magazine/article/December2016_CAN-Bus-Basics-Propeller-MCU**.

The Controller Area Network (CAN) began as a communication path for electronic equipment in vehicles. Unlike the RS-232 or RS-485 standards, the CAN standard (ISO 11898) provides electrical specifications and a format for information exchanges. Many newer microcontrollers include one or two CAN controllers that properly format data, set device addresses, provide hardware error detection, and identify communication errors if any. As a bonus, CAN based devices may operate in a multi-master network that uses more than one controller to supervise operations.

Datasheets and documents for CAN based microcontrollers might make them seem complicated to use. Thankfully, MCU manufacturers provide code libraries and examples that can get you off to a good start. You need not understand every aspect of the CAN bus and CAN controllers to communicate between devices. This article will help you learn enough to use basic bus communications in projects. Later articles will supply more details of CAN signals and protocols.

People who use RS-232 type communications understand how positive and negative voltages represent logic 1s or 0s. The CAN bus, on the other hand, uses the voltage *difference* between two wires — labeled CANH and CANL — to represent logic levels. The SN65HVD251 bus driver, for example, includes a *differential* driver and a *differential* receiver (**Figure 1**).

When you apply a logic 1 to the driver input (D) on an SN65HVD251 IC, the CANH and CANL outputs will have the same voltage level; about 2.5V for a 5V supply. We call this a *recessive* state on the bus. A differential receiver — including the one in the SN65HVD251 — would detect no voltage difference between CANH and CANL, so its receiver output (R) would give us a logic 1. Apply a logic 0 to the D input and the CANH output rises to about 4V, while the CANL output drops to about 1.4V to create a 2.6V *difference* between the two signals as shown in **Figure 2**. We call this the *dominant* state, and all differential receivers would produce a logic 0 at their R output.

Even if all other drivers produce recessive signals, a logic 0 input to

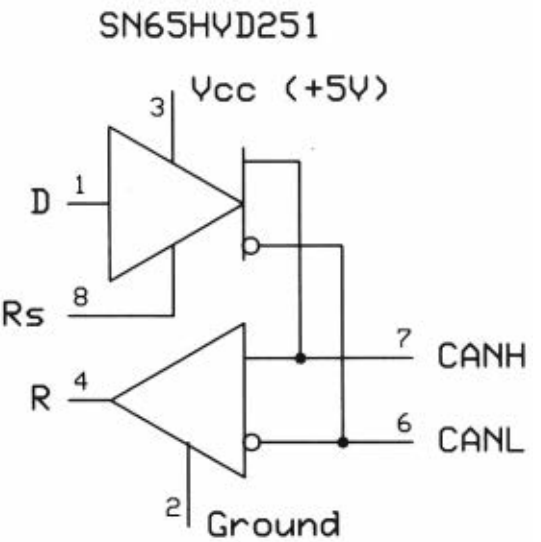


FIGURE 1. An SN65HVD251 provides a bus driver (top) and a bus receiver (bottom). The receiver output always indicates the state of the CAN bus. Ground the Rs input on this device, and do not make a connection to the reference-voltage output on pin 5 (not shown). A CAN bus does not need a common ground between devices (pin numbers for an eight-pin DIP).

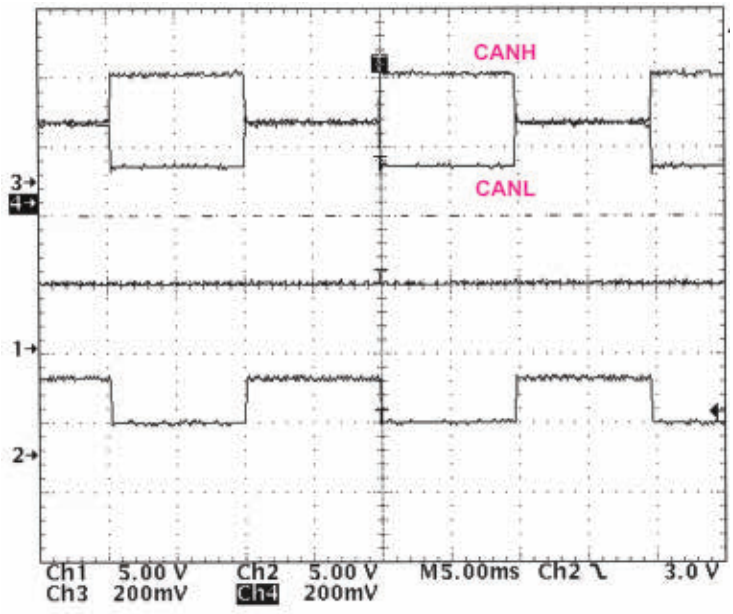


FIGURE 2. An example of a logic signal sent to an SN65HVD251 driver, and the CANH and CANL signals on a terminated bus cable. In this circuit, a logic 1 produces the recessive state in which CANH = CANL = 2.7V. A logic 0 forces the outputs into the dominant state where CANH = 4.1V and CANL = 1.4V for a 2.7V difference.

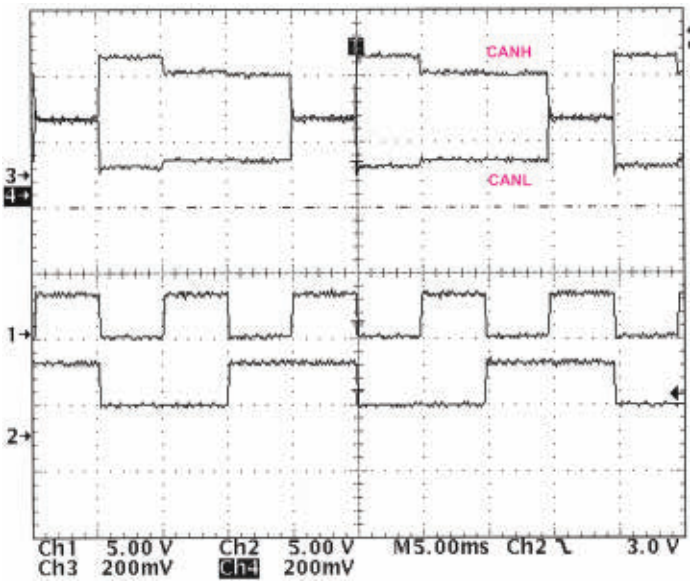


FIGURE 3. The top scope traces show superimposed CANH and CANL signals. The lower two traces show the input's signal applied to two drivers. Unlike "perfect" CAN bus signals shown in books and application notes, this figure shows measured results on a short twisted-pair bus.

any driver on the bus will always force the bus into the dominant state. Logic 0s always override logic 1s. **Figure 3** shows scope traces for the CANH and CANL signals on a bus with two square wave signals applied to two drivers. A logic 0 input sent to any bus driver input (D) forces the bus into its dominant state. Only when both drivers receive a logic 1 does the bus revert to the recessive state.

To operate properly and reduce signal reflections on a CAN bus, always use a termination resistor — typically 120 ohms — at each end (**Figure 4**). In parallel, these resistors closely match the bus-wire impedance of about 60 ohms. You can create branches off a CAN bus, but limit their length to a foot (0.3 m). As a rule, the longer a CAN bus is, the lower its data rate. For more information, see Note 1 in the **sidebar**. The scope images shown previously use only test signals, not a CAN formatted frame. Now, we leave the physical bus and explore how CAN communications work.

Identify and Arbitrate

A CAN frame begins with a logic 0 start-of-frame (SOF) bit followed by an 11-bit address that can identify 2,048 unique CAN devices. Networks won't use that many devices, but each manufacturer of CAN bus devices needs unique IDs for its products, so several thousand addresses make sense. The address — or identifier — also establishes a priority. Address 0 has the highest priority; address 2047 has the lowest.

You know logic 0s on the bus dominate logic 1s, so a logic 0 in an ID overrides a logic 1. Before a CAN controller starts to transmit, it monitors the bus and waits for a bus-idle condition (logic 1). Only then can it start to transmit a frame. What happens if two controllers start to transmit at the exact same time?

Each controller continues to monitor the bus to ensure its output matches what it detects on the bus. Consider the example shown in **Figure 5** when two devices start a frame simultaneously. Both begin a frame with SOF signals (**A**) and both detect a logic 0 from their respective receiver. So far, so good; so they continue to transmit. Then, the transmissions follow with a logic 1 most-significant ID bit (**B**) followed by a logic 0 (**C**). Again, the transmissions have no problems because the transmitted and detected bus values match the bit value sent from each transmitter.

At point **D**, the CAN1 TX output transmits a logic 0 and detects a logic 0 on the bus. However, the CAN2 controller detects a logic 0 on the bus as it starts to transmit a logic 1 (**Figure 5**, arrow). This mismatch causes the CAN2 controller to stop transmission because the CAN1 controller has started to transmit a higher priority ID. We call this process CAN-bus *arbitration*. Keep in mind the address — or ID — bits in the frame refer to the identification for the recipient device, not the address of the sending CAN controller.

Send the Data

Each CAN frame may include zero to eight bytes of data you can use for any purpose you choose. Bytes could include temperature values, actuator settings, switch-control bits, servo commands, and so on. This information goes to *all* CAN devices, but only the one with an 11-bit ID equal to the ID in the transmission will accept and save the data in a set of registers or memory locations. Some

devices such as commercial CAN based valves, sensor modules, and motor controls have factory-set IDs, or they could have DIP switches users can set for an address. Before your application software can respond to a specific address, you must load address registers — called filters — with the ID or IDs you choose for a device.

An eight-bit PIC18F66K80 MCU, for example, includes a CAN controller that holds six ID "filters," so you could create six IDs for the controller. Why have more than one address? A designer might assign an address for commands, another address for calibration values, and another address for I/O data. Before a controller saves any data bytes, the 11-bit ID in the received frame must match one of the IDs saved in a filter. CAN controllers also include 11-bit *masks* that let you ignore individual filter and ID bits. You would use mask bits if you want a CAN controller to accept addresses within a given range. A logic 0 in a mask means the controller ignores the corresponding filter bit and ID bit. A logic 1 in a mask bit forces the filter to exactly match the corresponding ID bit in a received frame.

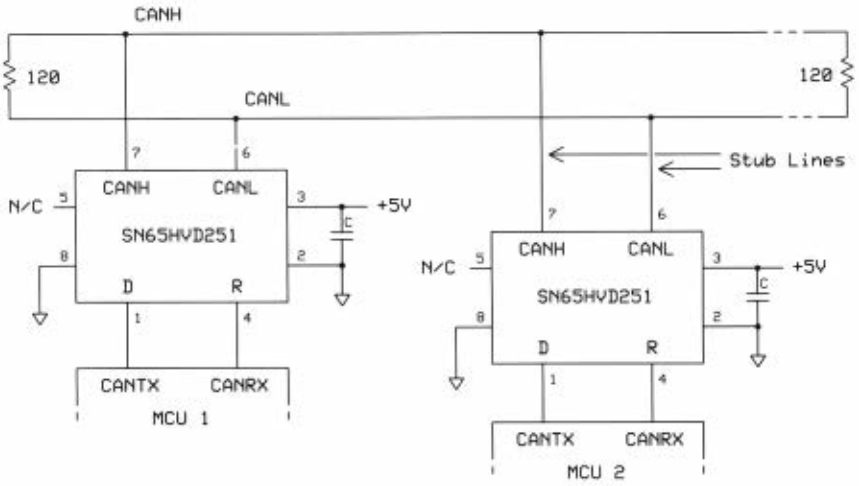


FIGURE 4. Circuit diagram for a portion of a CAN bus. Note the 120 ohm termination resistors at each end. Always limit stub lengths to one foot (0.3 m) or less.

Figure 6 provides an example for five bits. In examples **a** and **b**, the logic 1 mask bits do not affect the filter bits. The received ID in **a** matches the filter bits, so the CAN controller saves the received data. The **b** example shows a mismatch between the filter and the ID bits, so the controller saves no data. Example **c** shows a logic 0 at bit position ID0 in the mask, which means the corresponding filter bit gets ignored.

Either a logic 1 *or* a logic 0 at position ID0 in a frame's ID causes the controller to save the received data. If you had set both mask bits ID1 and ID0 to logic 0 (...11100_2), what ID *range* would the controller accept?

For the mask ...11100_2, the CAN controller could accept data in frames with four addresses: ...11100; ...11101; ...11110; and ...11111. Unless you need a range of continuous addresses for a controller, set all mask bits to logic 1 and set the CAN controller's ID filters as needed. (Revision 2 of the CAN standard provides for an additional 18 ID bits — a 29-bit ID — for over 500 million addresses. This type of addressing goes beyond the scope of this article.)

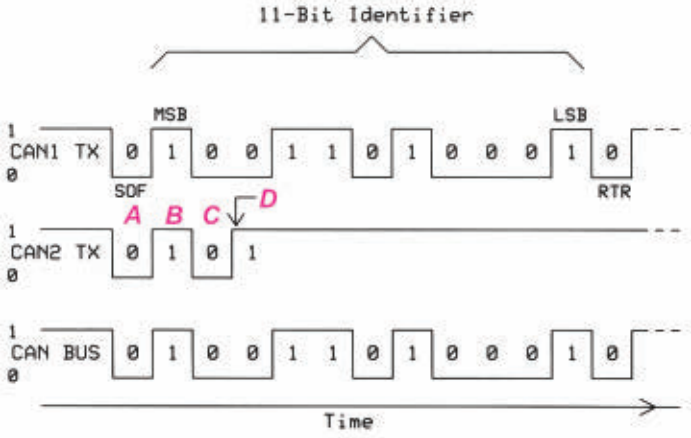


FIGURE 5. A timing diagram for the simultaneous transmission from the CAN1 controller addressed to a device with hexadecimal address 4D1 and from the CAN2 controller addressed to a device with ID 5A0. The frame sent from CAN1 has a higher priority (lower value), so it controls the bus.

Test CANbus Software

Now let's find out how to transmit and receive CAN frames. I used a Parallax Propeller MCU on a QuickStart board for the following examples because I'm familiar with the Propeller and with its Spin language. You can program it in C or assembly language too. The Propeller IC comprises eight identical processor cores (called cogs) that run independently. One or more cogs can handle CAN related tasks, while other cogs process results or prepare transmissions. Although the Propeller does not have a built-in CAN controller, software libraries offer standard CAN operations, and programmers have created code that works at data rates as high as 1 Mbit/sec.

The test circuit connected the CAN TX output (pin P24) to the CAN RX input (P25) on the same Propeller IC. A 10K ohm resistor from either pin to +3.3V completed the circuit (**Figure 7**). For a loopback test, you do not need a complete bus as shown earlier in **Figure 4**. (Many MCU suppliers connect CAN controllers to specific pins. The Propeller lets you assign any available pins you choose as CAN TX and CAN RX.)

The online free Propeller Object Exchange (OBEX) lists several CAN files with *methods* that send and receive CAN frames (go to **http://obex.parallax.com/object/745**). This folder includes a test program, "CANbus Loopback demo.spin" that transmits and receives data frames to test

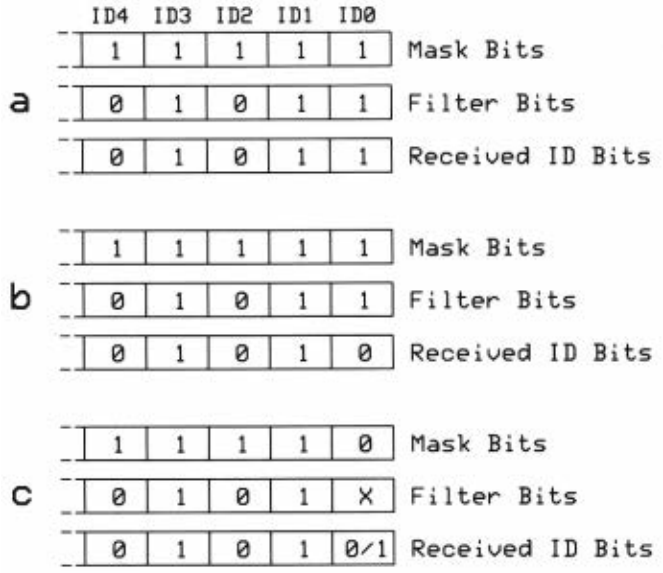


FIGURE 6. Three examples of how a mask bit affects the use of filter bits with a received ID. This diagram includes only the five least-significant bits (LSBs) of an 11-bit ID value. The bits not shown here operate the same way.

CANbus software. I found this program too complicated for novices, so you can use the simpler test, "CAN Loopback demo JT.spin." For an even simpler test, see "CAN Loopback demo JT2.spin." You can find both in the OBEX at **http://obex.parallax.com/object/845**. (Other MCU suppliers provide similar libraries of CAN software and code examples, although some get quite complicated.)

The free Propeller IDE (integrated development environment) — called the Propeller Tool — lets you open the CAN software, download it to a Propeller IC, run it, and modify it if you wish. The IDE includes the Parallax Serial Terminal (PST) that connects to the QuickStart board via a USB virtual port, which is also used to program the Propeller IC. When you run the CAN Loopback demo JT.spin program, the terminal window displays the information saved by the CAN receiver software (**Figure 8**). For more information about the Propeller IC and the Spin language, see the Propeller Manual, version. 1.2 (available on the Parallax website at **www.parallax.com**).

```
0F2  37
0F3  38 39
0F4  3A 3B 3C
0F5  3D 3E 3F 40
0F6  41 42 43 44 45
0F7  46 47 48 49 4A 4B
0F8  4C 4D 4E 4F 50 51 52
0F9  53 54 55 56 57 58 59 5A
0FA  5B
0FB  5C 5D
0FC  5E 5F 60
0FD  61 62 63 64
0FE  65 66 67 68 69
0FF  6A 6B 6C 6D 6E 6F
100  70 71 72 73 74 75 76
101  77 78 79 7A 7B 7C 7D 7E
```

FIGURE 8. This image displays three hex digits for the 11-bit CAN ID, followed by one to eight data bytes as seen in the Parallax Serial Terminal window.

How Does the Code Work?

First, a few words about the Spin language. When you need to use methods in an object file (one with a .spin suffix), you identify the object and then use that name as a prefix to the name of the method you want to use. The

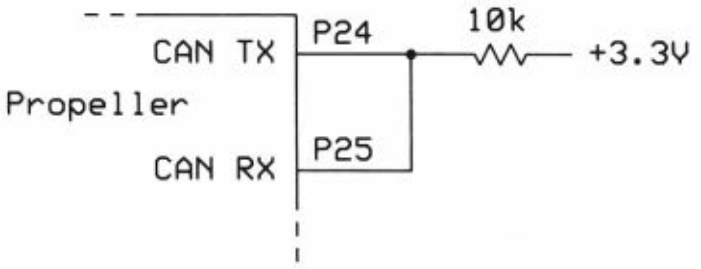


FIGURE 7. Circuit diagram for a loopback test of a CAN transmitter and receiver on a Propeller MCU IC. This type of test does not require a physical CAN bus with drivers and receivers.

Propeller "CANbus Writer 1Mbps.spin" and "CANbus reader 1Mbps.spin" object files include many CAN-control methods. To use them, give the object file a name:

```
OBJ
     writer : "CANbus writer 1Mbps"
     reader : "CANbus reader 1Mbps"
```

Then, you identify the methods in the object file this way:

```
writer.Start (25, 24, 1_000_000)
reader.loopback(true)
```

Now consider key methods that send and receive CAN frames. For more details, review the listings for the reader and writer object files noted above.

CAN Writer Methods

Located in object file: "CANbus writer 1Mbps.spin:"

1. `Start(rx_pin, tx_pin, bitrate)`
Define the CAN-TX and CAN-RX pins, and set the bit rate.

2. `Send(ident, bytes, d0, d1, d2, d3, d4, d5, d6, d7)`
Create and send a CAN frame with an ID value and as many as eight bytes of data.

3. `CheckReady`
This method returns a true condition if the transmitter-busy and error-detect bits are clear.

4. `Stop`
Stop CAN operations.

CAN Reader Methods

These methods provide a place to start when you create programs of your own. Keep in mind you must turn on and initialize the receiver and transmitter in any CAN based device you program. You cannot have "receive only" or "transmit only" devices on a CAN bus.

As I'll explain in a future article, the receiver plays an important role in a transmission. Located in object file

The Propeller Object Exchange (OBEX) lists several CAN files with methods that send and receive CAN frames; go to **http://obex.parallax.com/object/745**.

For an even simpler CAN test, visit **http://obex.parallax.com/object/845**.

For more information about the Propeller IC and the Spin language, see the Propeller Manual, version. 1.2 available on the Parallax website **www.parallax.com**.

CANbus reader 1Mbps.spin:

1. `Start(rx_pin, tx_pin, sync_pin, bitrate)`
Define the CAN-TX and CAN-RX pins, and set the bit rate. The sync I/O pin lets the two cogs synchronize operations via an I/O connection. Software can use the pin to communicate between tasks. If both cogs start successfully, this object returns the base address of the received data array. If the cogs do not start, the start method returns the false condition.

2. `ID`
Returns the Ident (ID) value, but does not wait for an ID value to appear in the receiver buffer memory. It returns the false condition if no ID was stored.

3. `DataLength`
Returns the number of data bytes in a CAN frame.

4. `ReadData`
Returns the contents of the data buffer, starting with the first byte. Successive use of this method returns the following bytes; second, third, and so on. It does not return "out-of-date" or old data. If the current data buffer contains only one byte, this method returns the value in byte 0, and false for the remaining bytes 1 through 7.

5. `DataAddress`
This method returns the address of a length-prefaced string that contains the data in the current receiver buffer.

6. `NextID`
Returns the Ident (ID) value in the next buffer, but it does not wait for an ID to appear. It clears the previous ID to zero. You use this method to advance to the next data buffer.

7. `SetFilters`
Set your mask value (if any) and as many as five filter values.

8. `Stop`
This method stops the CANbus reader 1Mbps object and frees the reader and the cogs for other uses.

What's Next?

A CAN controller has more capabilities than covered here. It will acknowledge receipt of a frame, identify frame errors, keep bit-timing clocks synchronized, and identify any defective CAN controllers on a bus. I'll discuss those and other topics in Part 2. **NV**

Notes

1. Corrigan, Steve, "Controller Area Network Physical Layer Requirements," SLLA270. Texas Instruments, 2008; **www.ti.com/lit/an/slla270/slla270.pdf**.

2. Voss, Wilfried, "A Comprehensible Guide to Controller Area Network," Copperhill Media, Greenfield, MA, 2008. ISBN: 978-0976511601; **http://copperhilltech.com/technical-literature**.

Replacing the 555 with a PIC

In 1984, Forrest Mims wrote a very popular booklet published by RadioShack: *Mini-Notebook 555 Circuits*. It contains 28 circuits using either a 555 or 556. All of the circuits are relatively simple, but they do form a very good basis for learning how to use the 555/556. The following article is the first of several which will detail the emulation of a 555 or 556 using a PIC. This particular implementation does both the monostable and astable modes, and is meant to be used only as a 555 or 556 emulator. Future articles will discuss the software needed to share the PIC's resources so that the emulation can be embedded within another application.

By Larry Cicchinelli

Post comments on this article and find any associated files and/or downloads at **www.nutsvolts.com/magazine/article/December2016_Replacing-555-with-PICs**.

PIC Based Enhanced 555

Since I retired in 2012, I have been volunteering technical assistance at a network of FM radio stations throughout the state. One of the issues they have had is that a device they use at several remote transmitter sites does not connect to the Internet router properly after a power outage. The station manager asked me if there was anything I could design that would help.

What I needed to build was basically a delayed one shot where the delay is retriggerable. The operation is such that if the audio stream coming from the target unit stops for some period of time, it will reset the unit.

The circuit and software I developed led me to this project: an "enhanced" multivibrator which I have used as a 555 replacement.

The features of the software are eight decades of time resolution, from 1 µs (range = 0) to 10 seconds (range = 7); each range has a span of 10 bits. For example, the 1 ms range spans from 1 ms to 1,023 ms in 1 ms steps.

Also, there are currently six operating modes:

1. One shot monostable multivibrator
2. Delayed one shot
3. Retriggerable one shot
4. Delayed and retriggerable one shot — both delay and pulse width are retriggerable
5. Astable multivibrator with separate control of both the on-time and the off-time
6. Same as 5, but with a synchronous gate

All of this is accomplished without having to re-program the PIC in any way. The only "programming" needed is to configure the PIC using up to four resistor dividers. By changing some definitions in the source code, the analog inputs may be swapped. However, RA3 and RA5 must be kept as shown.

RA3 (pin 4) has special internal handling due to its possible usage as the MCLR/VPP input. Its only other allowed use is as a digital input. RA5 cannot be used as an analog input. Due to the different pin functions, there is no way to make the device pin compatible with a 555. One obvious reason is that the 555 has ground on pin 1

and VDD on pin 8 — just the opposite of the PIC12F1572 I used.

The software is written so that both the mode and range values are read only once — at program startup. The pulse width and delay/off-time may be read before each pulse is generated.

Some of the advantages of this implementation as compared to a 555 or 556 are:

1. No capacitors
2. Very linear response for pulse width
3. Delayed pulse capability independent of the pulse width
4. Independent control of pulse width and off-time in astable operation
5. Very wide range of pulse width and off-time/delay
6. Output pulse width is independent of the triggering pulse width

Some disadvantages:

1. Shortest trigger to output delay is about 1 µs
2. Shortest pulse is about 3.3 µs
3. Longer minimum time between triggers due to reading the ADC (analog-to-digital converter) and other processing
4. Pulse width and delay use the same range

Some advantages of the 555:

1. Lower cost
2. Very low delay time between the trigger and the output pulse
3. Narrower pulses possible: <500 ns

As of this writing, these are the approximate prices for single pieces:

PIC12F1572 (DIP and SOIC)	= $0.60
NE555PSR (SOIC)	= $0.37
NE556DR (DIP)	= $0.45
CSS555 (SOIC)	= $1.55

Both the range and mode of operation have up to eight values. The operating values are determined by using the upper three bits of an analog voltage for each. The resistor network shown in **Figure 2** is an example of how to develop the voltages required for setting them. The voltage shown is for example only — you would never use +8V for the top of the divider! With the values shown, the divider draws 1 mA and the taps are 1V apart starting with 1.5V at tap 1.

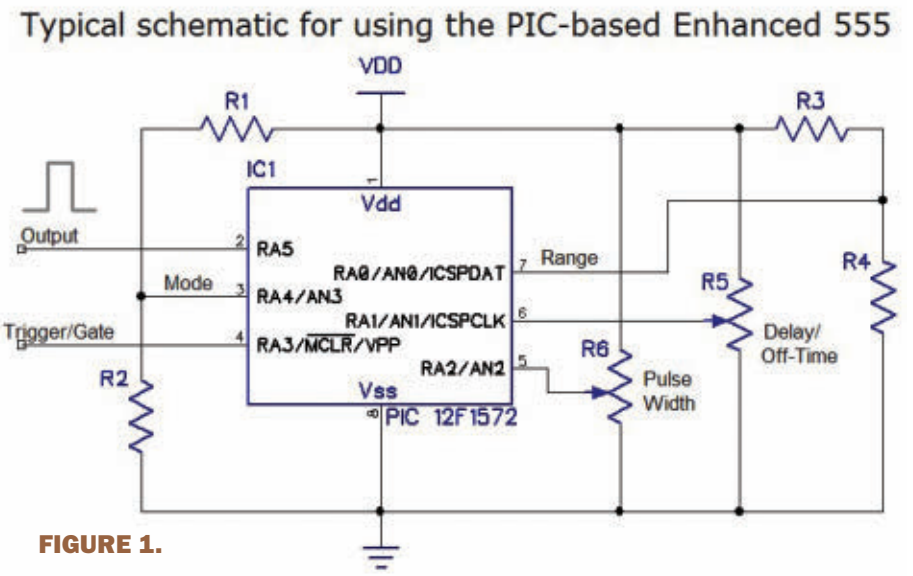


FIGURE 1.

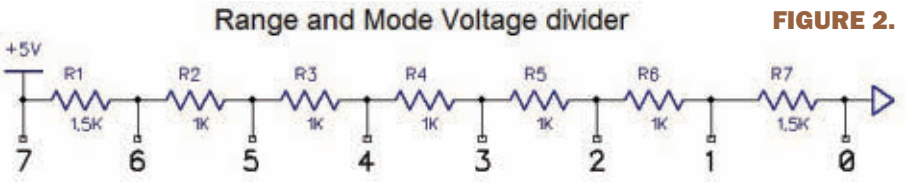


FIGURE 2.

The ADC in the PICs is ratiometric: The digitized value is the ratio of the applied voltage to the difference between the high and low references; it's then multiplied by 2^N, where N is the number of bits of the ADC. The references typically used are VDD and ground (the PIC12F1572 does not allow a choice for the negative reference). For this project, the formula is:

$$Value = \left(\frac{Vin}{PosRef - NegRef} \right) * 2^N$$

Using 8V for the positive reference, ground for the negative reference, and N = 10 for the 10-bit ADC, the formula reduces to:

$$Value = \frac{Vin * 1024}{8} \quad Vin * 128$$

For example: if Vin = 4V, I would expect the ADC to show a value of 1/2 full scale. Using the formula: Value = 4 * 128 = 512, which is 1/2 of full scale for a 10-bit ADC.

For this project, the program uses only the most significant three bits of the converted value for the range and mode inputs. The voltage divider is designed so that the six taps are exactly centered on the voltages needed for each of the six intermediate values. The voltages for the range and mode 0 and 7 use ground and VDD, respectively, and do not need a divider. Using range 2 as

an example, the voltage at tap 2 is 2.5V. The converted value (using the formula) is 2.5 * 128 = 320.

The 10-bit binary value of 320 is 01 0100 0000. As you can see, the upper three bits (bits 9-7), 010, are the value 2. Any value between 01 0000 0000 (256) and 01 0111 1111 (383) will also work. This yields a window of 127 or — with a 5V power supply — about 620 mV. By selecting the divider as I have, you can see that the voltage tap is centered in the window and yields the best solution, taking into account noise and circuit tolerances.

Since the ADC is ratiometric, the actual value of the positive reference is not relevant (keeping within the specifications of the PIC) as long as the input to the ADC does not exceed this value. The voltage divider resistors can stay the same as shown — also independent of the reference.

For a specific application, you may want to change the divider to only two or three resistors. For example, if you want range 3 and mode 1 (or range 1 and mode 3), the divider could be what you see in **Figure 3**. Notice that the total resistance is still 8K even though the positive reference is 5V. One parameter you need to take into consideration when selecting the resistor values is that the specifications for the ADC state that the maximum resistance of the source should be less than 10K ohms.

It is possible to extend this method beyond using only the upper three bits. Using the upper four bits will yield up to 16 possible "states." This will cut the noise/tolerance range in half to 63 counts of the ADC, or about 310 mV.

Theoretically, this method could be extended to the full number of ADC bits, but I would hesitate using it beyond six bits (a 68 mV window) mainly due to noise and other circuit tolerances. Keep in mind that six bits is about 1.5%, so using 1% resistors in the divider would be absolutely necessary. Even then, you would want to verify the voltage values at each tap.

Some Program Specifics

In order to get the best resolution for the ranges, the program changes the CPU clock speed depending on the range selected, as follows:

Ranges 0-4: 32 MHz clock = 8 MHz instruction rate = 125 ns period

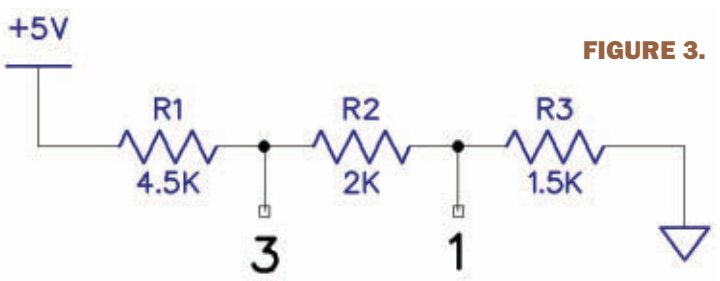


FIGURE 3.

Range 5: 8 MHz clock = 2 MHz instruction rate = 500 ns period

Ranges 6 and 7: 2 MHz clock = 0.5 MHz instruction rate = 2 µs period

The initialization of the processor starts the CPU with the 32 MHz CPU clock. Running at 32 MHz requires the use of the internal PLL which takes about 1 ms to stabilize. If range 5 is selected, the program will decrease the CPU clock to 8 MHz. If range 6 or 7 is selected, the CPU frequency will be lowered to 2 MHz.

These frequencies were selected in order to meet my goal of eight decades of resolution. Also, keep in mind that the PIC processors have four cycle instructions; for instance, a 32 MHz clock results in an instruction time of 125 ns.

Range 0 uses a timing loop with eight instructions so that the time to execute the loop is 1 µs. Ranges 1 through 7 use timer 1 with a preset value read from a table in program space. For all ranges — except 0 — the timer 1 clock is derived from $F_{CPU}/4$ with the prescaler value set to 8, which makes the timer 1 clock frequency = $F_{CPU}/32$.

The ADC reading for the pulse width and delay/off-time is used in software timing loops which — for ranges 1-7 — determine the number of times timer 1 times out.

There are several configuration macros which have been implemented to help a user who may want to customize the program:

DEBUG: Defining this value to 1 reassigns several of the I/O pins so that you can use the PIC debugger.

FAST: Define this value to 1 to speed up some of the times by reading the ADC only once during start-up. This means that not only the mode and ranges but the pulse width and delay time will also be fixed once the start-up code has completed.

AD_NO_BITS_0_1: Defining this to 1 causes the ADC *read* function to clear bits 0 and 1 of the reading. This will make the pulse width and delay/off-time values less susceptible to noise, at the expense of reducing the resolution by a factor of four while maintaining the same full scale capability.

AD_USE_8_BITS: Defining this to 1 causes the ADC *read* function to use only the upper eight bits of the conversion. This is different from *AD_NO_BITS_0_1* in that the resolution is unchanged, but the full scale capability of each range is reduced by a factor of four.

EDGE_REG: Define as *IOCAP* to trigger on positive edges; define as *IOCAN* to trigger on negative edges.

One of the ADC control registers is modified — based on the selected range — to make the sample

time/bit 1 µs (called TAD in the specs) which makes the conversion time (plus some processing) about 18 µs. The delayed one shot and astable multivibrator modes read the ADC twice: once for the delay/off-time and once for the pulse width. For the astable multivibrator mode, the reading for each state is taken immediately preceding that state so that both states are lengthened by the same amount of time.

For the one shot modes, the ADC is read before the program looks for the trigger signal so the reading times do not affect the trigger-to-pulse delay. However, the reading time does affect the minimum time between triggers.

I considered using interrupts to handle the ADC readings. However, since the interrupt can occur at almost any time, it would eventually happen when the timing loop was to terminate, causing the pulse to be delayed longer or the pulse width to be a little longer. The effect would be that some pulses would be longer than others. I felt that having the timing be more consistent was the better solution.

Something I noticed on the several PIC12F1572s I was using is that the ADC seldom returns a value of 0; even with the input grounded, it almost always reads a value of 1. This causes the pulse width and delay/off-time values to be one count larger than you would expect with a grounded ADC input. I thought about decrementing the values just prior to entering the timing loops, but decided against it since this would affect all settings.

All of the processor's "math" instructions operate on eight-bit values. However, it has auto-increment and auto-decrement instructions that use 16-bit pointers to access memory.

I decided to use one of these two registers as a 16-bit counter to contain the ADC value for the pulse width and delay. Even though this register accesses a memory location and copies it to the W register, I ignore it.

I use the auto-decrement mode each time through a loop and test the most significant bit. If it is on, the register has decremented to a negative value and the loop is terminated. Using this method saves three instructions which — in the case of 1 µs resolution — is significant. I did not need to use this method for the other ranges, but I chose to use it in order to be consistent.

The gated astable mode operates such that when the gate is enabled (high), the output pulse will be immediately generated. This will be followed by the off-time. The program will complete the entire cycle of pulse followed by off-time before it again looks for the gate.

The re-trigger pulse for the re-triggerable modes can happen any time during the cycle. The triggering edge of the re-trigger signal is captured by the processor and can be a very short pulse (>25 ns). If a re-trigger is detected, the current output cycle will be restarted. If the delayed one shot mode is selected, this means that both the delay time as well as the pulse width are re-triggerable. This is the mode I used in my application — as long as audio is present, it will re-trigger the delay time.

When audio stops for delay time seconds, the delay time will time out and a pulse will be generated. The pulse turns on a relay which removes power from the device for about 20 seconds and then re-applies power. The delay time must be long enough to allow the device to re-acquire the audio signal or the power-off cycle will repeat.

Some measurements for each of the ranges in mode 0 (one shot):

Range 0. Trigger delay: 870 ns–1.25 µs (three period window); PW for values 0-3 = 3.3 µs (26 periods).

All other pulse widths are about 160 ns wider than expected.

Ranges 1-4. Trigger delay: 880 ns–1.24 µs (three period window); PW @ 0V = 3 µs (24 periods).

All pulse widths are 3 µs (24 periods) wider than expected.

Range 5. Trigger delay: 3.4 µs–4.9 µs (three period window); PW @ 0V = 12.0 µs (24 periods).

For a non-0V PW value, there may be an additional 12 µs (calculated) for the pulse width.

Ranges 6 and 7. Trigger delay: 14.2 µs–19.6 µs (three period window); PW @ 0V = 48 µs (24 periods).

For a non-0V PW value, there may be an additional 48 µs (calculated) for the pulse width.

Some measurements for some of the ranges in mode 4 (multivibrator) are with the pulse width and delay/off-time inputs tied together. Ideally, this should yield a 50% duty cycle square wave:

Range 0. The shortest period with FAST = 0 is 42 µs. With FAST = 1, it is 8 µs. The narrowest pulse in either case is about 3.6 µs. The slight imbalance is due to processing time.

Range 1. The shortest period with FAST = 1 is about 27 µs. The narrowest pulse is 13.2 µs.

If anyone reading this article would like to experiment but does not have the capability to program PIC processors, I will be happy to program either a DIP or an SOIC. Just send me the IC and a SASE.

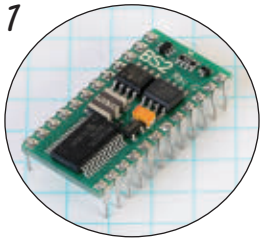
The source code for this project can be found at the article link as well as at my website (**www.qsl.net/k3pto**). The source code is well documented and should be easy to follow. Porting this code to another PIC processor should also be relatively easy to accomplish for those PICs that have at least 49 instructions. Some may need minor adjustments such as changing the RLF to RLCF. **NV**

Engineers Mini-Notebook 555 Circuits by Forrest M. Mims, III ©1984; also see **www.forrestmims.org**.

Name That Part!

Try this final photo quiz of 2016 and see how many parts you can identify. Some parts date back to the 1950s and earlier, while others can be found at your local RadioShack. For scale, the blue background grid contains 1/4 inch squares. Keep in the mind the photos have been sized to fit the layout. The correct answers can be found on page 64. Good luck!

1

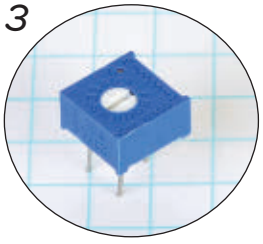


a. Arduino Micro
b. BASIC Stamp
c. Raspberry Pi 2

2

a. Coin Cell Battery
b. Super Capacitor
c. Alkaline Battery

3

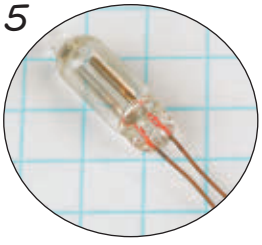


a. Trimmer Pot
b. Blue Dot Inductor
c. Variable Capacitor

4

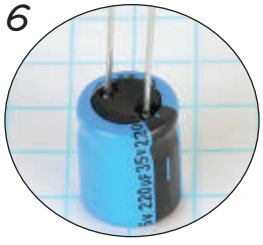
a. Panel Mount LED
b. N.O. Pushbutton
c. Fuse Holder

5



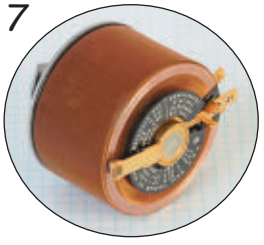
a. Krypton Bulb
b. Neon Bulb
c. Geiger Tube

6



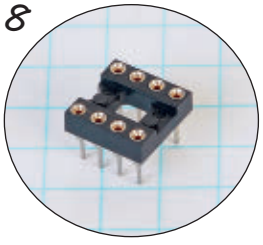
a. Mica Capacitor
b. Ceramic Cap
c. Electrolytic Cap

7



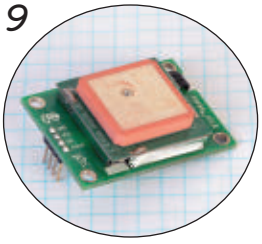
a. Variac
b. Triac
c. Varistor

8



a. DIP Socket
b. SIP Socket
c. Eight-pin D-Sub

9

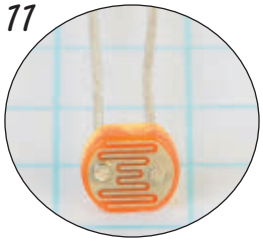


a. Bluetooth Module
b. GMRS Receiver
c. GPS Receiver

10

a. Mini Phone Jack
b. BNC Jack
c. Phone Jack

11



a. Touch Switch
b. Rain Sensor
c. Photo Resistor

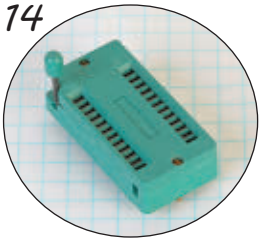
12

a. USB B Female
b. USB A Female
c. USB C Female

13

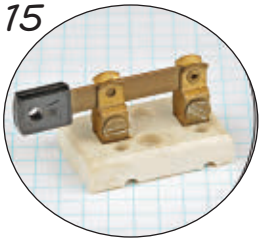
a. Triple NOR
b. Quad NAND
c. Hex Inverter

14



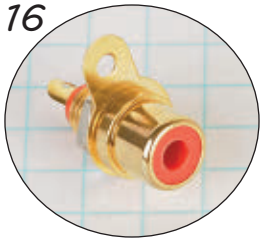
a. ZAP Socket
b. ZIF Socket
c. ZAF Socket

15



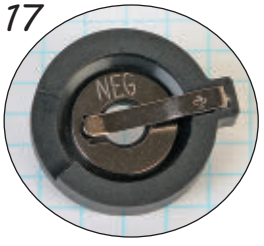
a. Knife Switch
b. Toggle Switch
c. GFCI Breaker

16



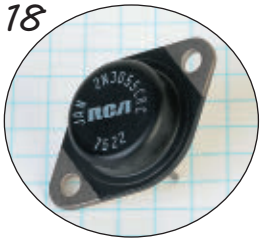
a. RCA Jack
b. QVC Jack
c. Power Jack

17



a. Battery Holder
b. Battery Charger
c. NEG Heatsink

18



a. TO-220
b. TO-3
c. TO-92

Presented by David Goodsell

Scoring: 0-7 = Novice 8-15 = Good 16-18 = Expert

Celebrating 46 Years!

CHANEY ELECTRONICS, INC.

Order Online at: www.chaneyelectronics.com

All Orders Must Include Shipping

Toll Free: 1-800-227-7312
Fax: (480) 451-9495
PO BOX 4116, SCOTTSDALE, AZ 85261

Compact 3.2V to 30VDC 3-Digit Red Voltmeter

- Compact 3-digit voltmeter
- Voltages from 3.2VDC up to 30VDC.
- Size only 0.55" X 0.9" (excluding mounting ears).
- Simple to use just connect red to (+) and black to (-). Requires about 23mA.

C12002 $3.49 ea. LIMIT 5

Large Digit VOM

- Compact VOM is only 5" X 2.74" X 0.92" Thick. 3½ digit VOM.
- Has 5 ranges for DC volts 0-1000V, DC Amps 0-10Amp in 4 selectable ranges,
- AC volts 0-750V in 2 selectable ranges, resistance 0-2 megohms
- In 5 selectable ranges: diode test, continuity buzzer test, NPN, PNP transistor hFE test position and square wave output.
- Complete with test leads in a sealed retail blister package.

G21263 $5.95 ea. LIMIT 5

Small 2V-24V to 5V-28V Boost Module

- Operates on 2VDC up to 24VDC
- Boost the output to 5VDC up to 28VDC at 2Amps max.
- Size only 0.65" X 1.5" X 0.25 Thick.
- Has trimmer potentiometer to set output voltage.
- Boost a low voltage solar panel of 3V up to 14V to charge a 12V battery or power a 12V project.

G21500 $4.00 ea. LIMIT 5

5 Chip 220,000MCD Warm White 10mm LED

Powerful 10mm 5 chip white LED produces 220,000MCD features 2500K-3500K color temperature, 40 degree viewing angle, 3.2 to 3.6VDC forward voltage and 100mA forward current. Brand new factory fresh!

C18242 $1.49 ea. LIMIT 5

4-40Vin- 1.25-37Vout Step Down Converter with Meter

Compact step down converter can operate on input voltages of 4VDC up to 40VDC and has a variable trimmer to adjust output from 1.25V - 37V. Output current rating is 2Amp continuous. Has buttons to switch display so that it can indicate input or output voltage (has a ± 0.5V tolerance on meter indication). Size only 2.8" X 1.56" W. Uses LM2596 IC regulator.

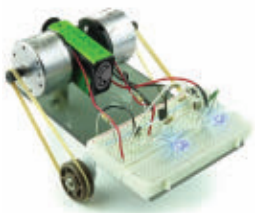
G21272 $6.95 ea. LIMIT 5

33 in 1 Deluxe Exploration Lab

Educational breadboard lab incorporates an introduction to Ohm's Law, electromagnet circuitry, motor circuitry, transistor circuitry, IC circuitry, and 7 segment display circuitry. Build 33 electronic experiments & activities.

C6709 $27.50 ea.

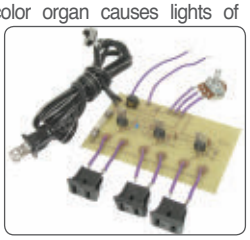
7 in 1 Reusable 9VDC Robotic Lab



- Educational fun and inexpensive to build electronic robots.
- You get an acrylic base, breadboard, motors and electronic components to build 7 different robot activities by reusing the components.
- Great first robotic kit for students without previous electronic knowledge.
- Operates from one 9V battery (not included) .
- Size of robot about 5.25" X 3.25" X 2.75" H.

C7093 $24.95 ea.

120VAC 3 Channel Color Organ Kit



Very popular 3 channel color organ causes lights of your choice to flash to the beat and frequency of the music. Simply connect the color organ to your stereo speaker. Kit features level control and 3 separate AC outlets to connect Christmas lights, lamps, etc. Operates from standard 120VAC. Skill level 2. Requires Soldering.

C4530 $13.25 ea.

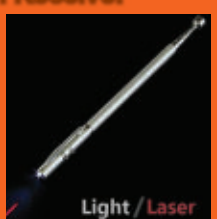
The Light Spider Robot II Kit

Our new and improved Light Spider Robot II Kit can now be used with any LED or Incandescent Flashlight! Using its electronic brain to find or follow a light source, this amazing robot quickly scoots across any smooth, hard surface. Skill level 2. Requires Soldering & 9V battery.

C7049 $13.25 ea.

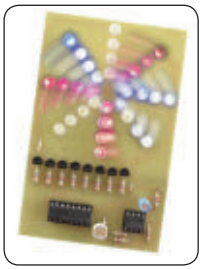
Merchandise Orders of $20.00 or More Can Receive:

FREE!
3 in 1 Telescopic Magnetic Pick-up Tool



Simply Add Stock#
NVTOOL
To Your Cart
Before Checkout
LIMIT 1 coupon per Order

Magic Light Sweep Kit (Red/White/Blue)



This kit produces sweeping red, white and blue light bars that magically speed up or down depending on the ambient light level. Makes a great attention getting display. Size of board: 2.6" x 4.1". Operates from one 9V battery (not included). Complete with all parts, PC board and instructions. Skill level 2. Requires Soldering.

C6439RWB $12.25 ea.

Learn To Solder Robot Kit

This unique flashing robot will not only help you learn to solder but will also be the center of attention wherever it is placed! The 3 1/2" tall x 7/8" wide. Requires one 9V battery. Skill Level 1. Requires Soldering.

C6845 $9.50 ea.

Insanity Alarm Kit

Set this kit in an inconspicuous place with light shining on it and it remains perfectly silent, however, as soon as the light is turned off the kit emits a high pitched irritating tone! Skill Level 1. Requires Soldering & 9V battery.

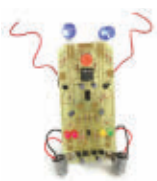
C6240 $5.65 ea.

120VAC Variable Strobe Light Kit

Features variable flash rate and straight xenon strobe tube. Speed can be varied from about 120 to 200 flashes per minute. Great for discos, parties, dances, displays, etc. Operates from standard 120VAC. Size of board: 3" x 2". Skill Level 2. Requires Soldering.

C3071 $11.50 ea.

Electric Slider Learn to Solder Robot Kit



Scooting Robot requires one 9V battery and uses 2 powerful motors and a 1 IC circuit to slide in a weird pattern on any smooth surface. Skill level 1. Requires Soldering.

C7012 $10.25 ea.

Electronic Cricket Kit

An electronic cricket that starts "chirping" every time the lights go out. If you turn on the lights to look for the "cricket", it becomes silent again. Uses a CMOS IC and CDS cell circuit with an adjustable "cricket " sound control, that enables to adjust your cricket to the chirp that you find most irritating. Size of the PC board: 1 7/8'' x 2 1/2''. Skill Level 1. Requires Soldering & 9V battery.

C6707 $5.45 ea.

Computer Control and Interfacing with the NI MyRIO

By David Ward

Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/magazine/article/December2016_Computer-Control-with-NI-MyRIO.

National Instruments' MyRIO places dual-core ARM® Cortex™-A9 real time processing and Xilinx FPGA customizable I/O into the hands of novices. With its onboard devices, seamless software experience, and library of courseware and tutorials, NI MyRIO provides an affordable tool that can be used to do real engineering.

What computer geek would not jump at the opportunity to "control the real world" with their computer and develop their own programs to do this? There are many circuits, kits, systems, etc., available today to accomplish just that. One of the very best units available for computer interfacing and data acquisition is the NI MyRIO-1900 unit (shown in the photo above). The NI MyRIO units cost about $500. However, when all of the NI MyRIO features are examined, it will become clear what a powerful computer interfacing device it can be.

There are several reasons why the NI MyRIO comes out on top when compared to other less expensive computer interfacing systems. First, there's the hardware configuration and the many available I/O options. The second big MyRIO advantage is that it is programmed in NI's LabVIEW graphical programming language, which is included with the MyRIO unit. (By the way, LabVIEW is an acronym which stands for Laboratory Virtual Instrument Engineering Workbench.)

LabVIEW is programmed in a language which NI calls "G" for graphical. National Instruments is a world leader in computer control and data acquisition hardware and software. Learning how to use LabVIEW and their hardware can add an important skill set to one's resume.

The MyRIO can be controlled, as well as communicate in several different ways: through a Wi-Fi connection; through a USB cable; and finally, a program can be embedded into it and run all on its own without a host computer. It can also send out and receive data

Part 1: An introduction to the National Instruments' MyRIO and LabVIEW software. How to build and deploy your first LabVIEW VI (virtual instrument) into the MyRIO.

FIGURE 1.

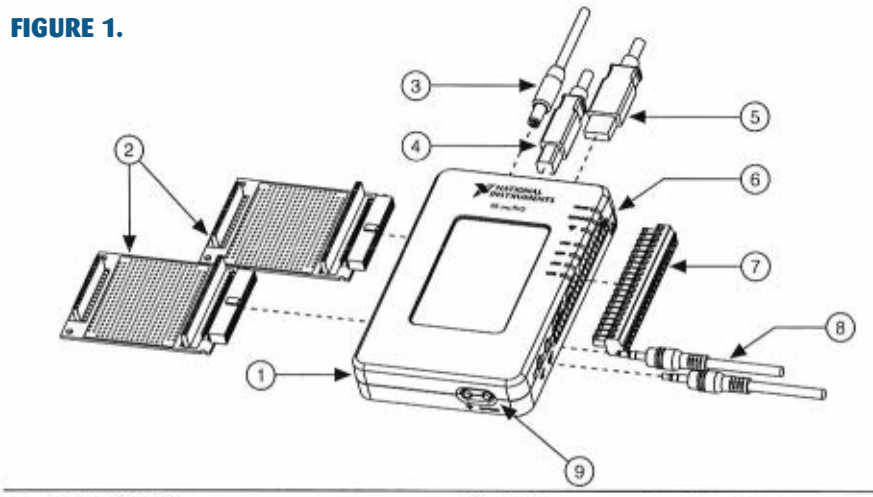
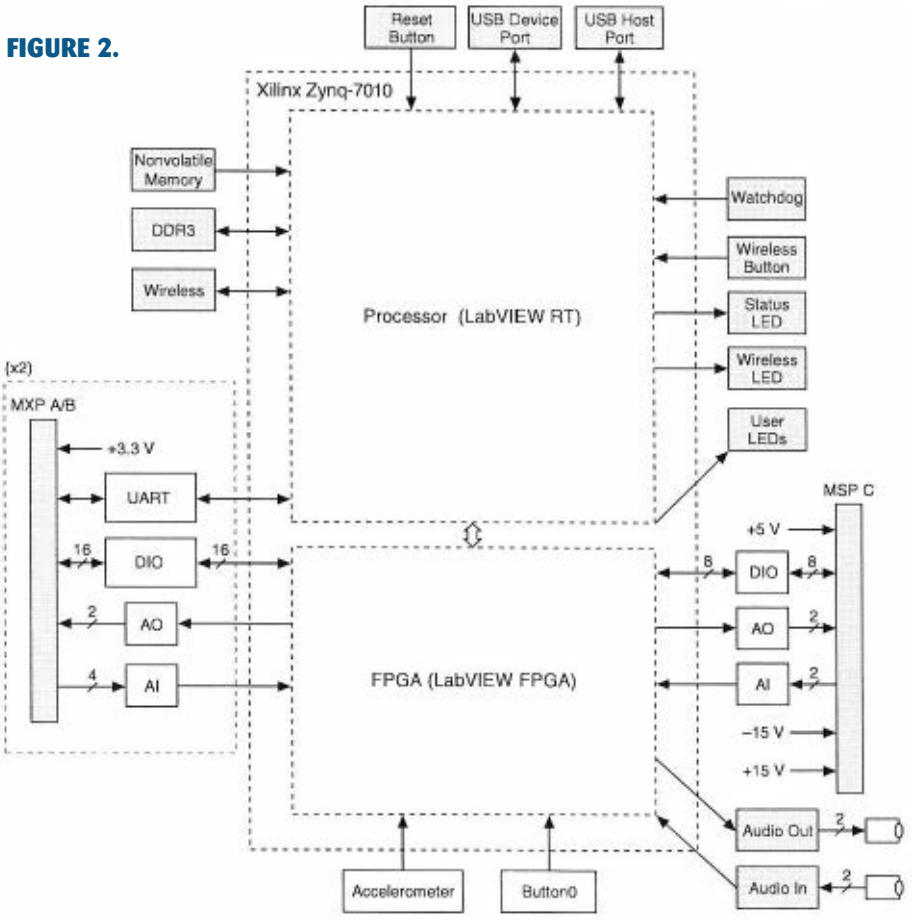


FIGURE 2.



wirelessly as a W-iFi router. This last feature allows the user to control the MyRIO with an Apple iPad running NI's Data Dashboard which is available for free from the iTunes store, as well as for Android tablets from the Google Play Store.

The MyRIO hardware is a block approximately 5-1/2 inches tall, 3-1/2 inches wide, and 1" deep (**Figures 1** and **2**). The unit's brain is a Xilinx Zynq-7010 FPGA. Although programming FPGAs (field programmable gate arrays) can be a daunting task, LabVIEW makes it quite simple to deal with, and the user generally does not need to be concerned with the details of how the FPGA operates. Advanced users can get more involved with the FPGA if they desire.

The unit has three expansion ports available: two MXP ports (MyRIO expansion port) called A and B; and one MSP port (mini system port) called C. The unit comes with a protoboard that can connect to ports A and B, and a screw terminal block that can be connected to port C. Each A and B port has 16 digital I/O lines and port C has eight for a total of 40 digital I/O lines.

Ports A and B have two analog outputs each and port C has two as well, for a total of six analog outputs. Ports A and B have four analog inputs and port C has two, for a total of 10 analog inputs. Ports A and B each have a UART (universal asynchronous receiver transmitter) port for serial communications with such things as LCDs (liquid crystal displays), and can emulate several serial communication protocols such as I^2C, SPI, and RS-232. It has audio in and audio out connections. It also features a user programmable pushbutton on the bottom and four user programmable LEDs, so programs can be run that require no external connections whatsoever (other than the DC power source). It has a DC power connector and comes with an AC adapter to 12 VDC. However, the unit can be powered

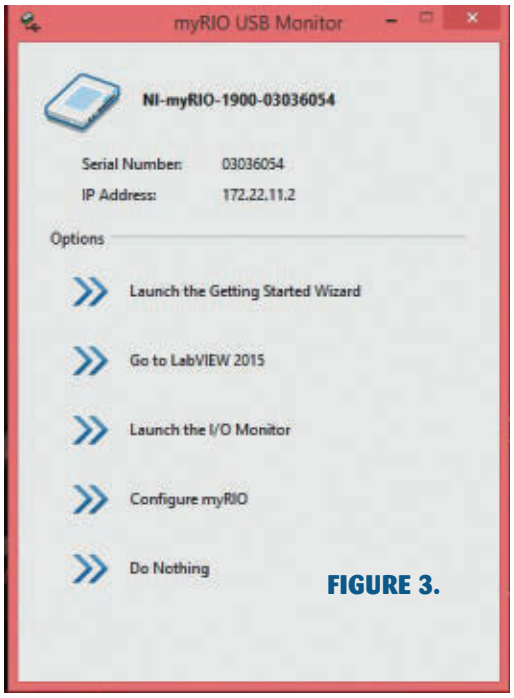


FIGURE 3.

FIGURE 4.

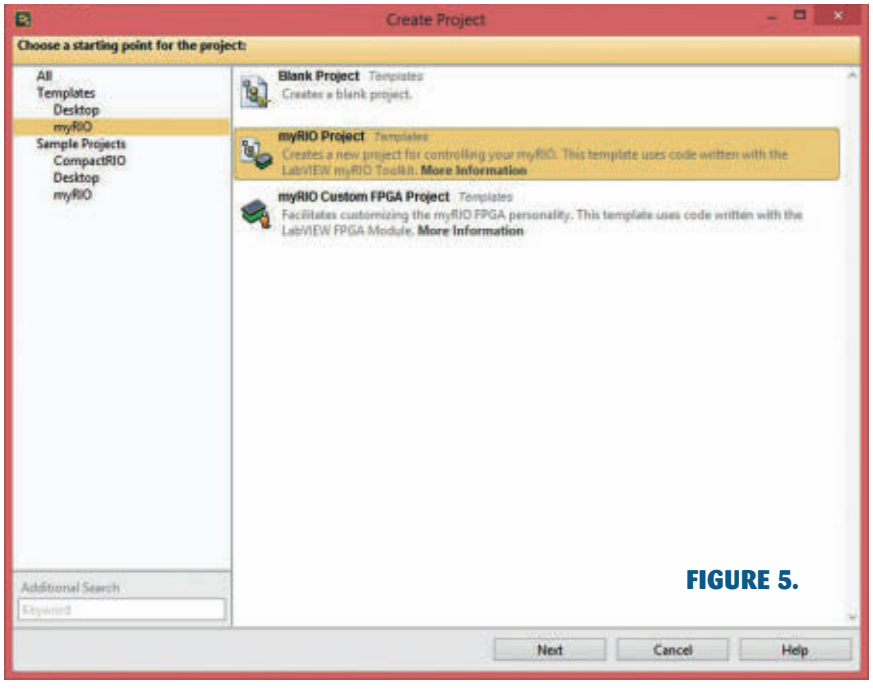


FIGURE 5.

by DC from as low as 6V up to 16V.

It comes with a USB cable and a port for communicating and programming by a host PC. It has a USB host port that can also support a USB memory device, which is handy for data logging applications. It has a Wi-Fi interface for wireless control and communication, and the unit has an internal three-axis accelerometer as well.

The LabVIEW software that this article will demonstrate is LabVIEW 2015 running on a Windows 8 PC. It will probably be best to see how LabVIEW works through a demonstration. Before attempting the following demonstration program, it is assumed that the user has installed NI LabVIEW 2015 and has connected the MyRIO unit to the host PC through the USB cable. When this is done, NI software will automatically detect the presence of the MyRIO unit and make sure that the proper software is downloaded and installed into the unit. The NI programs will also give the user a chance to test the pushbutton, LEDs, and accelerometer, so you can be sure everything is working correctly before you build your own program.

This first demonstration program will be as simple as possible and not use any external connections. The program will simply light up one of the four built-in LEDs when the user pushbutton (BUTTON0) is pressed. It's located on the bottom side of the unit.

LabVIEW can be launched either from a Windows menu or — each time the MyRIO is plugged into a USB port — the myRIO USB monitor will pop up and give the user five options. Select the second option: Go to LabVIEW 2015 (**Figure 3**). After launching LabVIEW, you will see what's shown in **Figure 4**. From here, select Create Project. You should now see the window shown in **Figure 5**. Next, select myRIO > myRIO Project and you will see the window in **Figure 6**, and then finally check out the project view window in **Figure 7**.

This project has a program ready to run called Main.vi; expand the + next to the NI-myRIO-1900. Double-click on Main.vi and you should see the front panel shown in **Figure 8**.

From this front panel window, select Window > Show

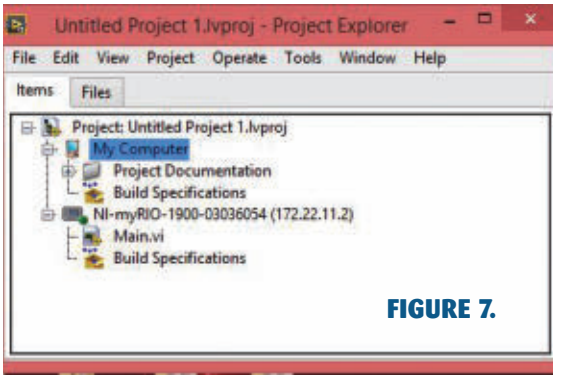


FIGURE 7.

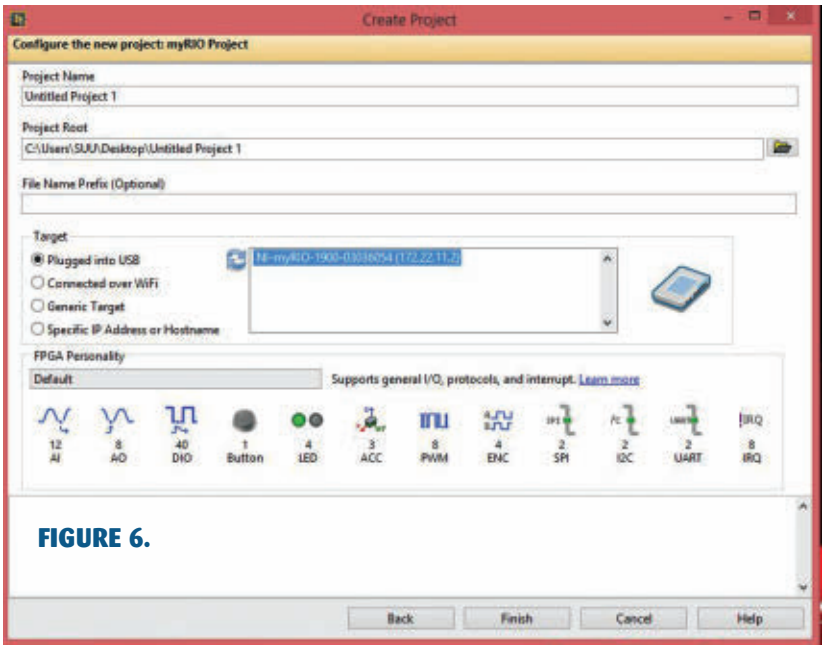


FIGURE 6.

Block Diagram and you should see the block diagram for this project as in **Figure 9**.

So, a LabVIEW project will consist of at least three different windows: the Project View window; the Front Panel; and the Block Diagram. The front panel is the user interface that will allow control of the program, as well as display information to the user. Normally, users won't need to see the block diagram unless they are the one developing the program. The block diagram is where the actual "code" or "instructions" for the program are located. You normally don't actually "write" lines of code in LabVIEW but rather graphically place objects, wire them together, and set their properties to behave how you want them in order to satisfy your program's requirements.

LabVIEW programs are called VIs or virtual instruments, and always consist of at least one front panel and one block diagram. You can click on the "RUN" arrow or icon on the top of either the front panel or block diagram to compile, deploy, and run the VI on the MyRIO unit. This Main.vi will chart the three-axis accelerometer as you move the unit around in different directions.

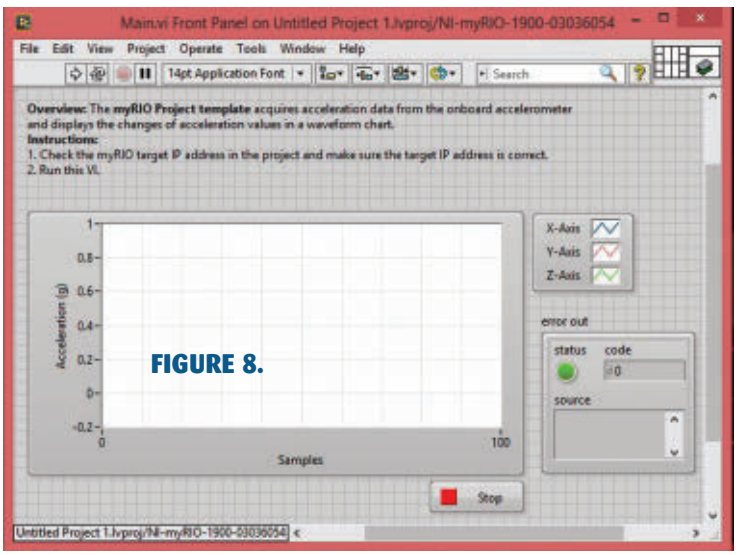


FIGURE 8.

Let's make our own VI to turn one of the four built-in LEDs on when BUTTON0 is pressed. From the Project View window, left-click on NI-MyRIO, then right-click to get the pull-down options as shown in **Figure 10** — New > VI. You should now see a new blank front panel and blank block diagram. Looking at the Program View window, your new VI should be called "Untitled 1." If you go to either the front panel or the block diagram, you can select File > Save As and rename the VI to something else, such as demo_1.vi; see **Figures 11** and **12**.

When first using LabVIEW, it is helpful to have several pop-up

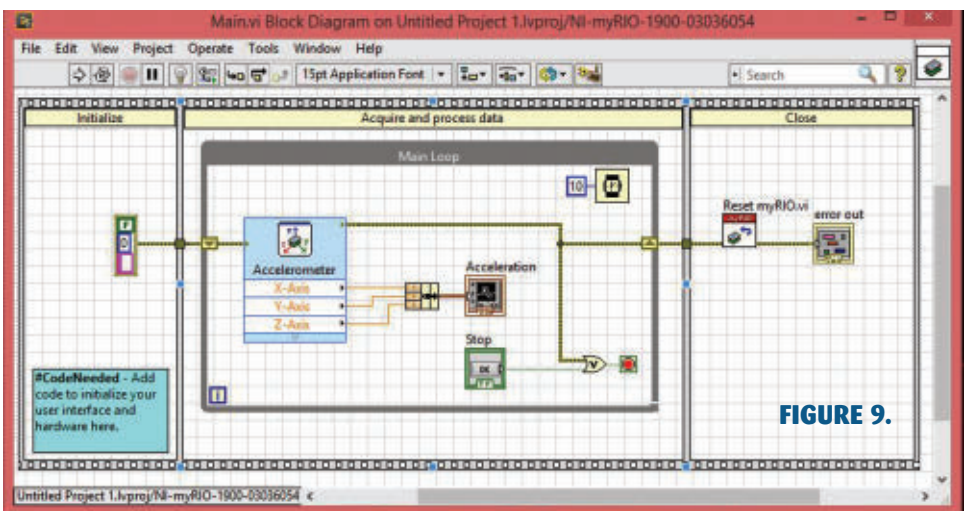


FIGURE 9.

items available to refer to. After using LabVIEW, you can generally hide these windows as they take up a lot of the work space. Select View > Functions Palette and View > Tools Palette on the block diagram, and View > Controls Palette and View > Tools Palette on the front panel. By selecting Help > Show Context Help, you will be shown a help window as your cursor hovers over most items on the front panel and the block diagram; refer to **Figure 13**.

The palettes you select objects from to place on your VIs are called the Controls palette on the front panel and the Functions palette on the block diagram. Nearly everything you place on either the front panel or the block diagram will have a corresponding icon on the other window.

In order to have the VI run continuously on its own, place a "While Loop" on the block diagram as shown in **Figure 14**: Functions > Programming > Structures > While Loop. You may notice that the Run arrow icon appears "broken." This is because the "Stop" icon in the lower corner of the While Loop is unconnected. By clicking on the broken Run arrow icon, you will be prompted or shown where the error is.

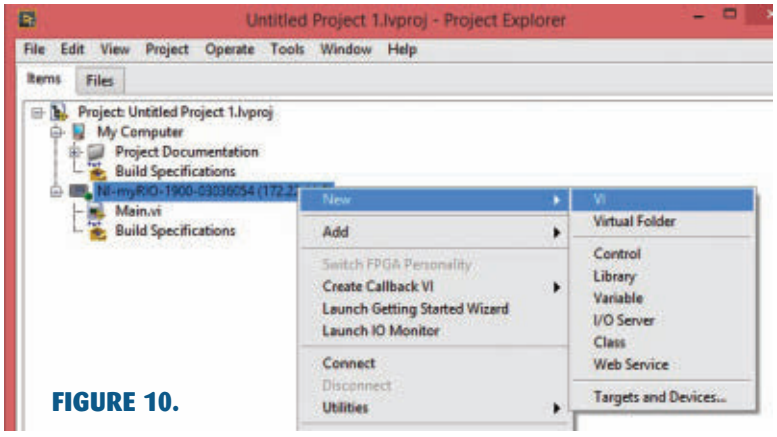


FIGURE 10.

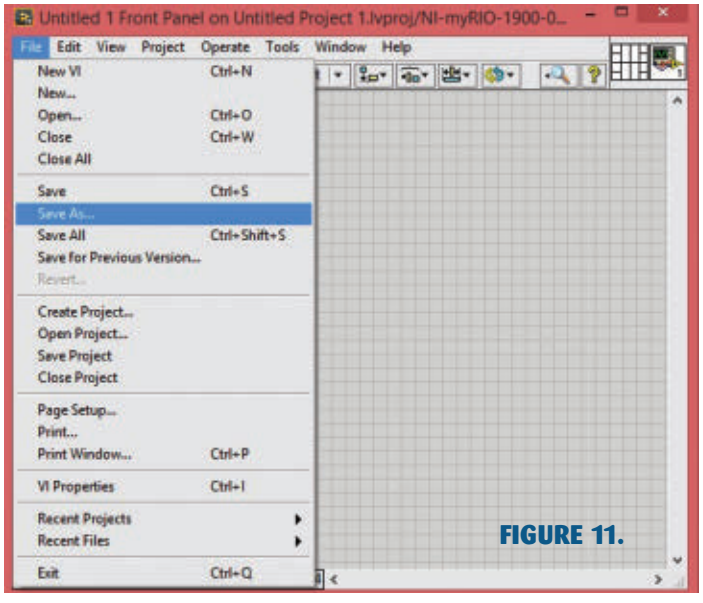


FIGURE 11.

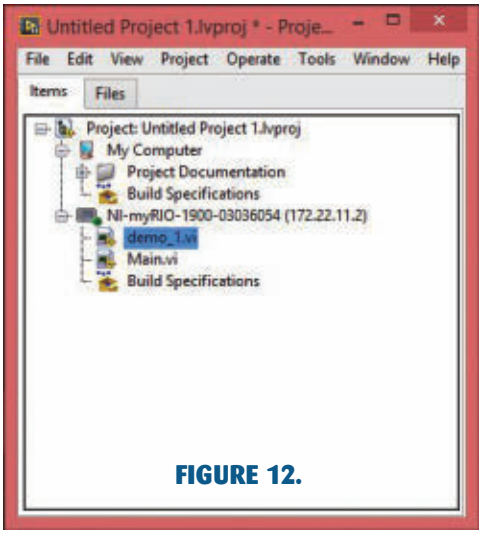


FIGURE 12.

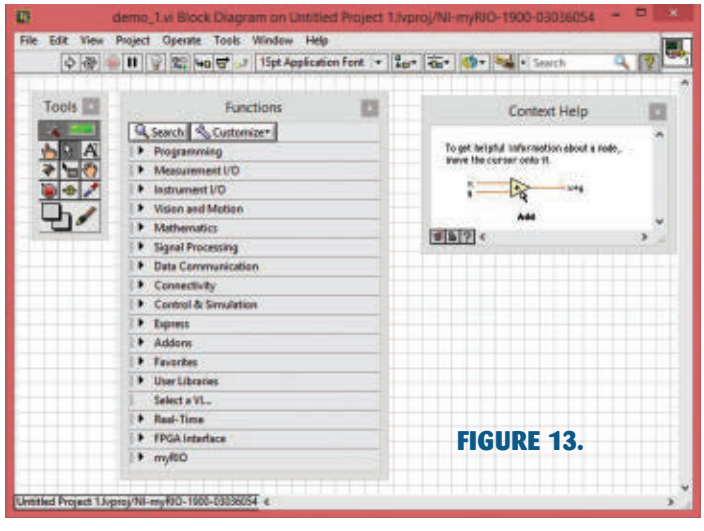


FIGURE 13.

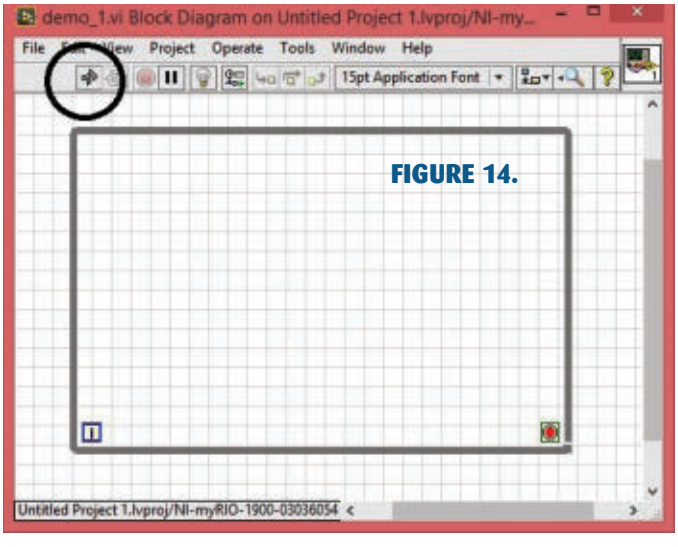


FIGURE 14.

This can be corrected by selecting the While Loop Stop icon. Then, right-clicking from the pull-down menu, select Create Control (see **Figure 15**). If you look at the front panel window, you will see a corresponding Stop button that the user can click on to stop or terminate the VI when it is running (**Figure 16**). This should have also "fixed" the broken Run arrow icon as well.

Now, let's place a control for the button and the LED inside of the While Loop on the block diagram. From the Functions Palette on the block diagram, select and place a button and LED inside the While Loop: myRIO > Button and myRIO > LED (as shown in **Figure 17**). While placing the LED function down, you should have seen the window in **Figure 18** which allows you to select which LEDs you want activated. This demonstration unselected all except LED0. The Run icon should be broken now because these

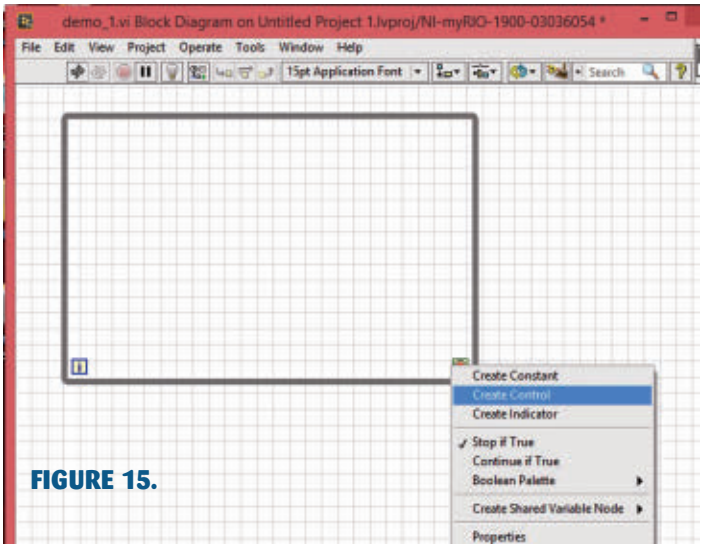


FIGURE 15.

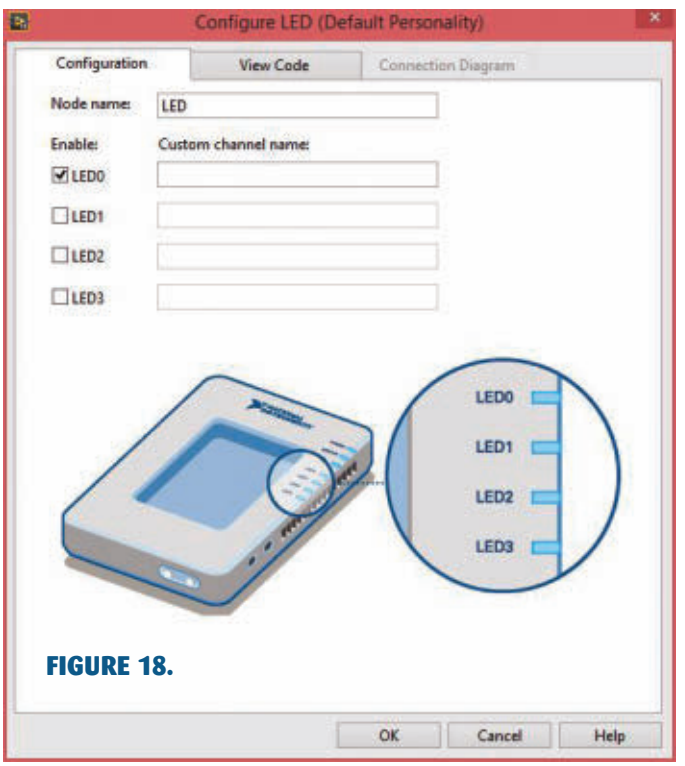


FIGURE 18.

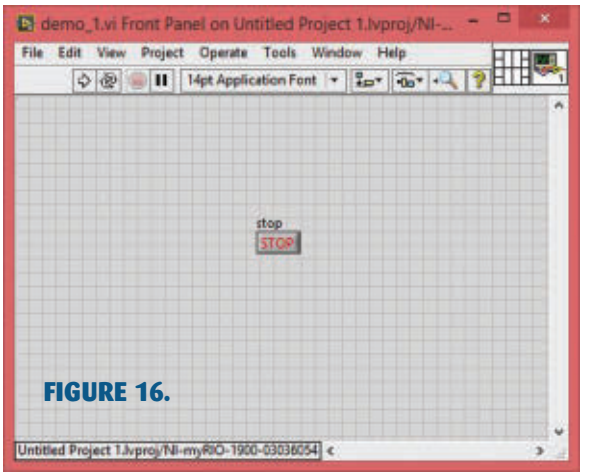


FIGURE 16.

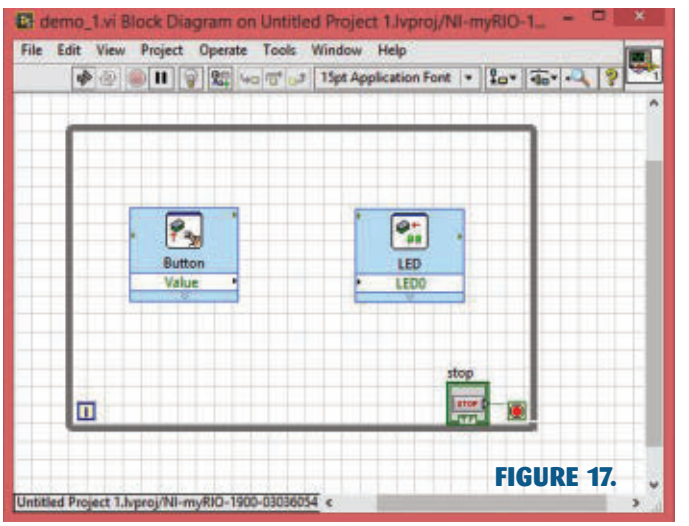


FIGURE 17.

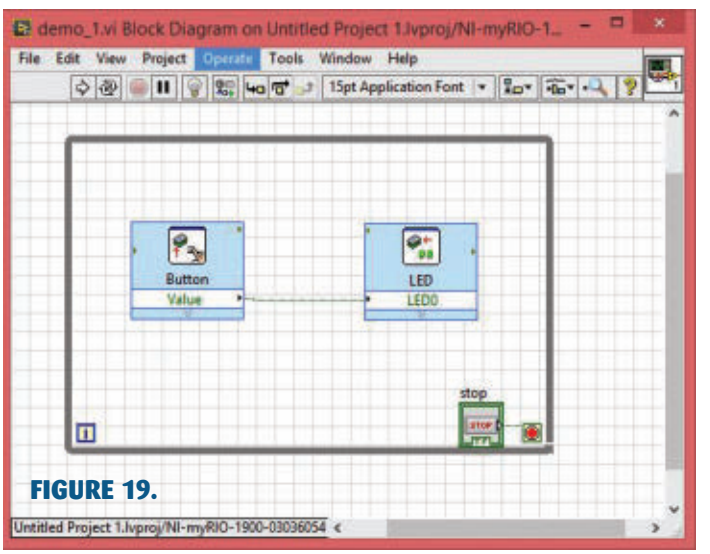


FIGURE 19.

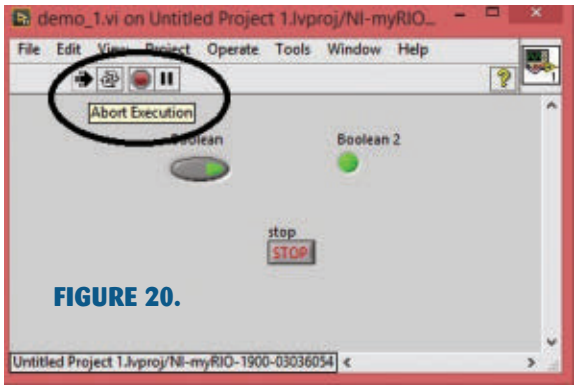


FIGURE 20.

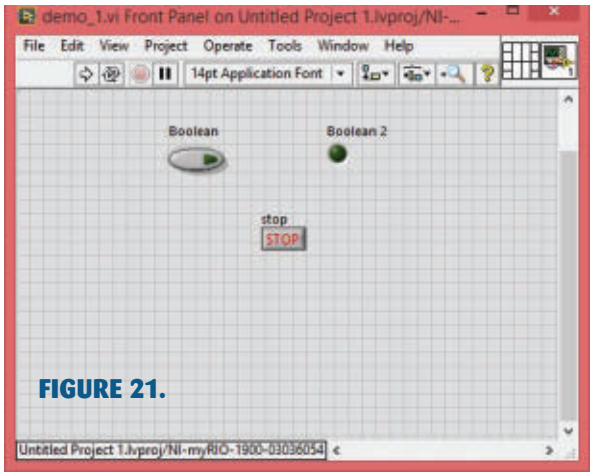


FIGURE 21.

two functions have not been "wired" together. As your cursor comes near the "value" terminals on the two Function icons, it should change from a cross cursor to a "connect wire" cursor (which looks like a roll of wire), so the two can be wired together. Refer to **Figure 19**. This should fix the broken Run icon so the VI can be run.

Click on the Run icon and when BUTTON0 on the bottom of the MyRIO is pressed, the blue LED0 should light up. You will see a couple of windows while the program is compiling and deploying, and you will be prompted to save the VI as well.

Clicking on the Stop button on the front panel will stop the program. You may have also noticed a red "stop sign" icon to the right of the Run icon; this will also terminate a program — especially if it is caught in an endless loop somewhere — but it is not the recommended way to terminate or stop a normally functioning VI (**Figure 20**). Using the stop sign or Abort Execution icon to stop a VI can cause problems if your VI needs to complete tasks as it closes, such as logging data or turning certain I/Os off or on at the conclusion of the VI's operation; this will be demonstrated later on.

Hopefully, your first VI was successful. The front panel or user interface for this demo VI is very bare and not very interesting to look at. Let's add a virtual pushbutton and a virtual LED to the front panel so that either they "or" the actual BUTTON0 and LED0 will operate the VI. On the front panel from the Controls Palette > Boolean > pushbutton, place a pushbutton and LED on the front panel: Controls Palette > Boolean > round LED; see **Figure 21**. Wire these two items together on the block diagram along with an "Or gate" as shown in **Figure 22**. The Or function can be found on the block diagram from Functions Palette > Boolean > Or.

The previous connection between the BUTTON0 and LED0 will need to be deleted before the new connections can be made. As your cursor hovers over the connection, right-click and then from the pull-down menu select "Delete Wire Branch" (see **Figure 23**). Now, when the VI is run, clicking on either the virtual button on the front panel or the physical BUTTON0 on the MyRIO unit will turn on the physical LED0 and the virtual LED on the front panel. You might also note that if the front panel pushbutton is in the "on" position when the VI is stopped, LED0 will remain on.

It is usually a good idea to design your VIs so that

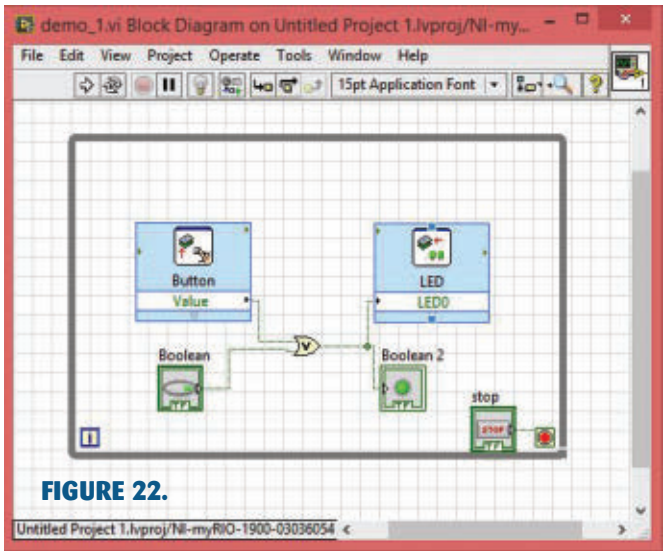


FIGURE 22.

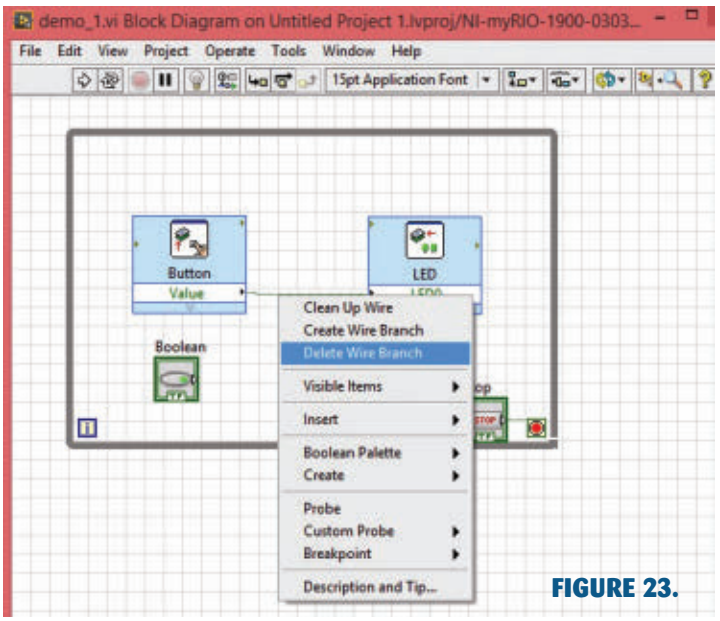


FIGURE 23.

they turn the I/Os off when exiting a Run condition. Without this "shutting down" of the VI, you will have to run it again and click the front panel pushbutton to the off position before stopping the VI.

To make your VI shut off LED0 when stopping the program, place a flat sequence around the While Loop on the block diagram as shown in **Figure 24**: Functions Palette > Programming > Structures > Flat Sequence. Click on the flat sequence frame, right-click, and then from the pull-down menu, select "Add Frame After" as shown in **Figure 25**. In the blank frame to the right of the While Loop frame, copy and paste the LED icon as shown in **Figure 26** (ctrl C > ctrl V will work for copying and pasting).

On the value terminal, click and add a constant which should be an "F" for false or to turn the LED off after the While Loop is terminated. If you click on the F, it will change to a "T" for true (but we need it to be F for now). Now when the VI is run — even if the virtual pushbutton on the front panel has turned LED0 on — it will turn off if the stop button on the VI's front panel is pressed.

However, if you click on the Abort Execution stop sign to halt the VI's execution, it will stop while it is in the While Loop and never complete the final frame of the VI (which turns LED0 off). This is a good demonstration of why it is best to stop VIs with an internal control rather than use the Abort Execution icon. This should only be used to stop a VI which cannot be stopped in any other way.

Being able to use many of the NI MyRIO's abilities is going to take more than one article to complete. Therefore, this will be a series of five articles covering different features of the NI MyRIO and LabVIEW software, building up to a final temperature control system with data logging capabilities. Part 2 will cover building and deploying an embedded VI and controlling the MyRIO wirelessly with an Apple iPad, so stay tuned!

These five articles cannot cover all of the MyRIO's features, but will give the reader a solid starting place to build upon. Hopefully, you have been able to duplicate all of the demonstrations shown in this article.

The NI website at **www.ni.com** contains many tutorials and other helpful items for all of their hardware and software products. You may want to spend some time looking over their many available online resources. **NV**

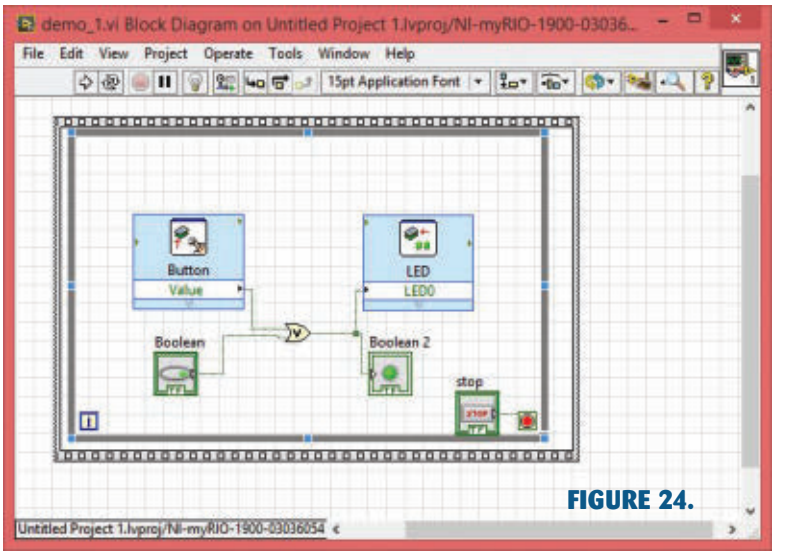


FIGURE 24.

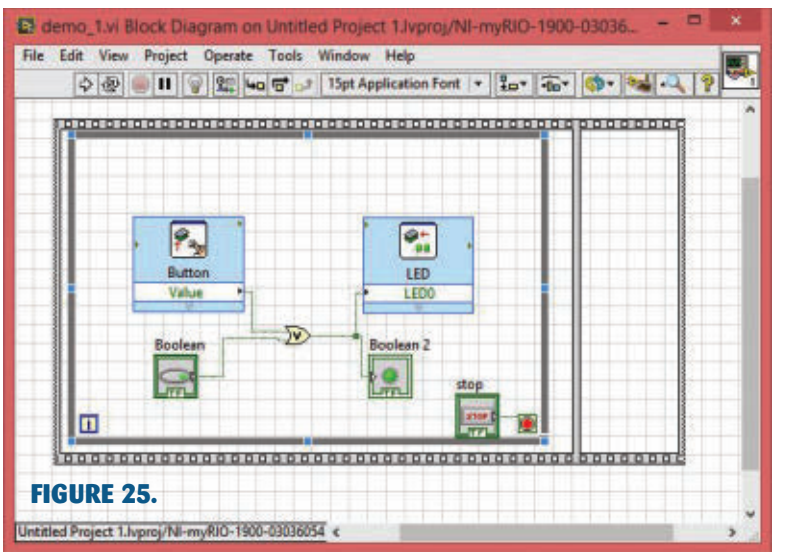


FIGURE 25.

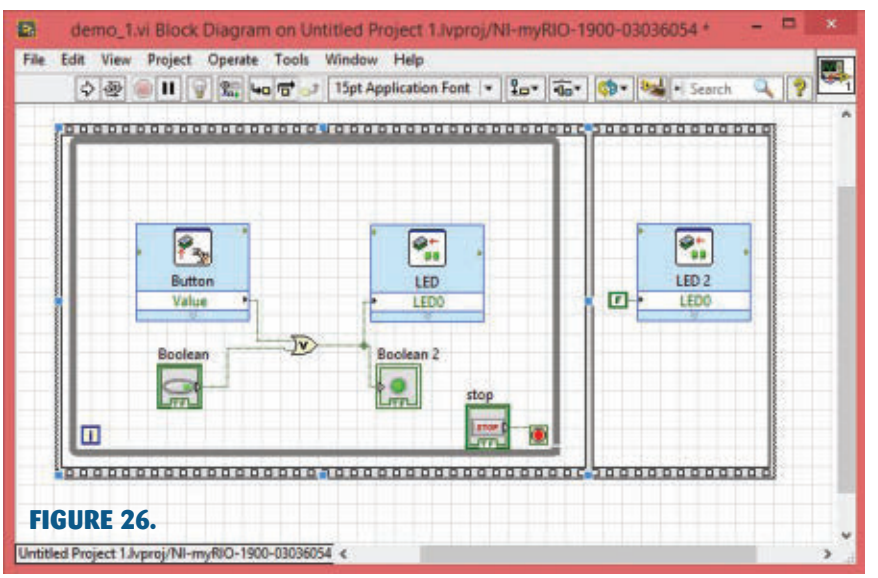


FIGURE 26.

■ BY CHUCK HELLEBUYCK

Using 3D printers for practical projects on your workbench.

3D Printed On-Air Flashing Sign

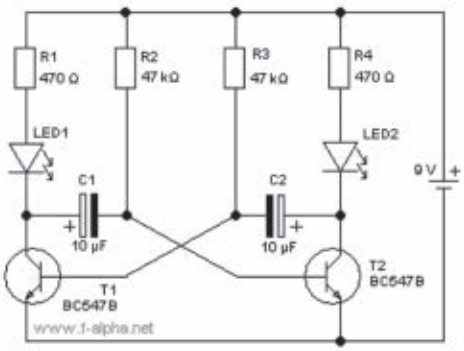
When I record my YouTube videos, I need to be left alone without interruptions. I do most of my filming in my garage workshop and this can sometimes be a problem. Occasionally, one of my family members will enter the garage and oftentimes mess up a film clip. It's unintentional, but I wanted a way to let them know I was filming. I tried a sign on the door but that didn't always catch their eye. So, I decided to design and 3D print a portable sign that has flashing LEDs to bring attention to the fact I'm filming, so don't enter. The finished design is shown in Figure 1.

■ **FIGURE 1.** On Air sign.

The design was created from scratch using Tinkercad 3D design software. It's so simple to use since it was originally designed for kids, but still has many features that make it powerful enough for creating 3D printed prototypes. For the electronics, I decided to design my own circuit board and router it out on my X-Carve CNC.

Electronics

The first step was to get the circuit to work. I did this on a breadboard. I found a simple dual LED flasher circuit also known as an astable multivibrator (**Figure 2**) that I found at **f-alpha.net**, so no microcontroller was needed or involved. I guess you could call it old school electronics.



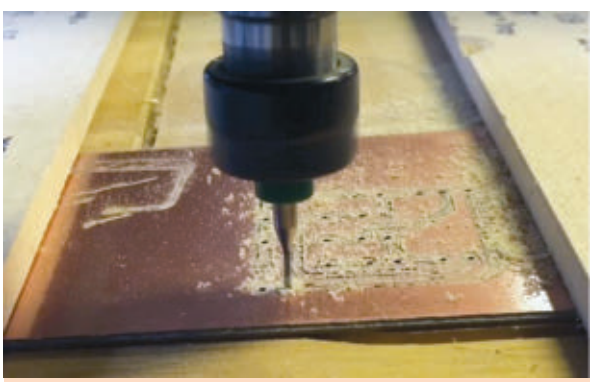
■ **FIGURE 2.** LED flasher.

The cost of the parts would be less than $1 but since I had most of them in my component drawers (some of which are 3D printed), I was able to build a prototype rather quickly.

From there, I created a board layout in ExpressPCB since I know that software so well. Rather than spend money on a single prototype, I decided to use my CNC to produce the board. I imported the final design into Copper Connection board layout software (that I'm still learning).

Copper Connection can produce Gerber files from any ExpressPCB file, so I imported the ExpressPCB file, exported the Gerber files, and then imported those into Flatcam software.

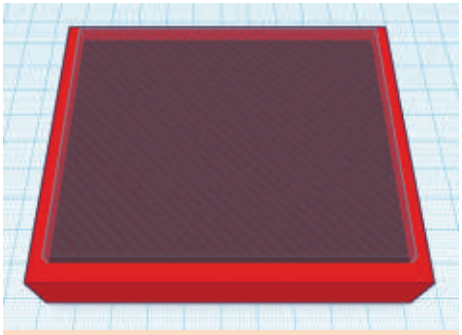
Flatcam is free software that can create the G-Code for my CNC. Flatcam will produce a G-Code file for routing out the traces and also the drill file for creating the holes. I used my X-Carve CNC to do the dirty work on a single-sided bare copper board (**Figure 3**).



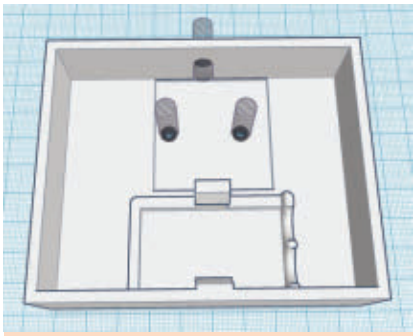
■ **FIGURE 3.** CNC'ing the circuit board.

With a little bit of soldering, the flashing LED board was ready to use (**Figure 4**). It runs directly off a nine volt battery so no regulator is required (again, no micro makes this easier). The 3D print design could now be created because I had my final board size and component placement. I needed two holes in the box for the red LEDs to poke through. I also wanted an on/off toggle switch so a hole on the top of the box was called for, plus I needed a nine volt battery holder of some kind.

Post comments on this article and find any associated files and/or downloads at **www.nutsvolts.com/magazine/article/December2016_Practical3DPrinting**.



■ **FIGURE 5.** Hollow box.



■ **FIGURE 6.** Holes in the box before grouping.

■ **FIGURE 7.** Final Tinkercad design.

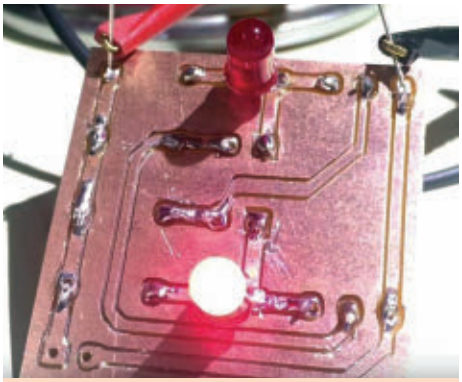
3D Design

The design was completed in **Tinkercad.com** 3D Design software that is in the cloud. All you need is a browser like Chrome or Firefox to create the design, and then you can export the .stl file. I started with a box element.

The box can be set to exact dimensions using the optional ruler tool. When I click on the box with the ruler tool included, the dimensions appear and I can manually change them with the keyboard. This allows me to make exact size adjustments. I chose a box 100 mm x 80 mm.

A second box element that is slightly smaller is placed inside the first box, except the second box is made into a hole. When the two are combined, the hole takes away material leaving a hollow box behind. The hole box is positioned to leave a 3 mm wall all the way around. Even the face of the box is 3 mm thick.

The next part was the battery holder. I found a nine volt battery holder already designed on **Thingiverse.com**. I imported the .stl file of the battery holder into the design using the Tinkercad import feature. Then, I placed the battery holder inside the box near the bottom and grouped them together. This merged the two into one design.



■ **FIGURE 4.** Assembled flashing LED .

Putting holes in a 3D print is not just simply a matter of measuring the part and making a hole the same size. When the layers of plastic are placed on top of each other to build the object, the plastic is squished. This means that holes tend to come out smaller than expected. This has to be done properly at the time of the 3D design phase.

If you want the hole to fit tight, I've found that a 0.5 mm larger diameter works well. To get a clean but loose fit, then plan on a hole 1 mm bigger than what size your hole is actually intended to be. I placed the holes in their proper place along with a recessed square to accept the circuit board.

I would later realize that the LEDs prevented the board from sitting flat, so I had to use double-sided foam tape to fill the gap while also holding the board in place.

Holes are made by using the cylinder hole object set to a diameter to match the intended hole size (plus the 1 mm) and inserting it into the design at the proper location. Then, when the object and the cylinder hole are combined, a hole is left in the box.

The finished box design is flipped over and has the words ON AIR recessed in the front. This was done with the Text object, which is also converted into a hole and placed into the box surface. When the box and letters are combined, they form a recessed ON AIR in stencil style (which I selected).

3D Print

The finished 3D file is then exported from Tinkercad as a .stl file and loaded into a slicing programming to produce the 3D printer G-Code (it all comes down to G-Code). I like to use Simplify3D, which is a $150 software package, but you can get free slicers such as Cura or Slic3R and produce similar results.

The slicer is where all the 3D printing settings are adjusted. I chose to use a 0.3 mm layer height which doesn't produce as smooth of a print, but it cuts the print time significantly. So, for a first run prototype, some people like to use 0.3 mm and then finish it with a final 0.1 or 0.2 mm layer height.

The box walls can be made with different infill (or the amount of plastic insulation) in the walls. I chose a 25% infill, which is done in a crisscross pattern. If this box needed to be very strong, then a 50-100% fill would be the better option.

I decided to print this one on

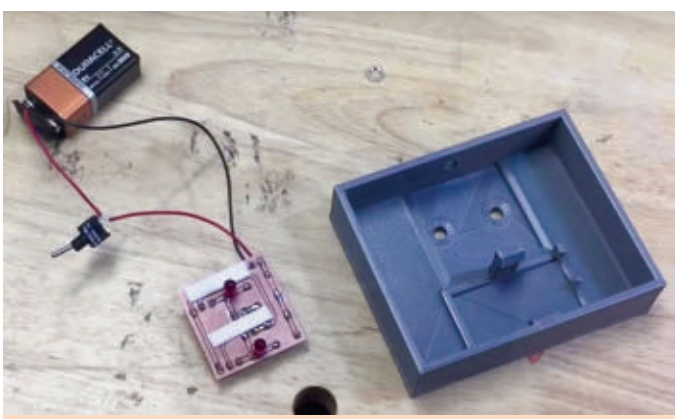
■ **FIGURE 8.** FLASHFORGE Dreamer 3D printer.

my FLASHFORGE Dreamer 3D printer. This design would have fit on my smaller MP Select Mini 3D printer, but I was printing with ABS plastic. I wanted ABS because it is more flexible than a PLA plastic, which is better for the battery clip.

Printing ABS works best in an enclosed chamber to keep the heat in. This reduces the risk of splitting or warping of the print, which ABS tends to do. So, the FLASHFORGE Dreamer is the better choice with its enclosed chamber.

The design printed well and was ready to install the electronics into. I connected and soldered a nine volt battery cable to the board, then cut the positive red wire and installed an SPDT switch in series. The circuit and box are shown in **Figure 9** ready to assemble.

I used a couple strips of double-sided foam stick tape to hold the board in place. I could have designed screw holes and posts in the plastic, but since it was a first attempt, this was a bit easier. The nine volt battery snaps into the clips at the bottom. The switch inserts into the hole at the top of the box and is held in place by a nut that screws on the switch threads. That's it.



■ **FIGURE 9.** Ready to assemble.

Resources

Check out my website and blog:
www.elproducts.com

My YouTube channel:
www.filamentfriday.com

My 3D designs:
www.thingiverse.com/elproducts/designs

Tinkercad:
www.tinkercad.com

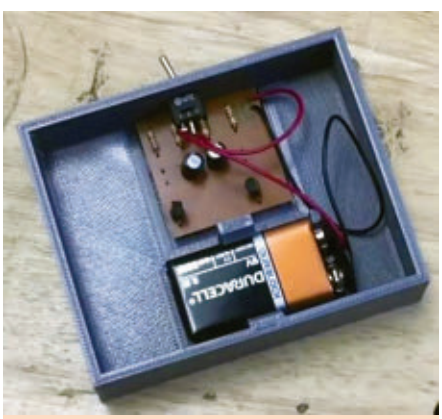
The design is complete and a flip of the switch produces a back and forth flashing light indicating I'm filming my next YouTube clips. Placing this in a window of my shop or hanging it from the door indicates that filming is going on.

Conclusion

This is not the most complex design electrically or mechanically, but it does show a step-by-step approach to creating an electronically based product with the help of a 3D printer. This is exactly why I bought my first printer, so I could make custom cases in the way I wanted rather than getting a box off the shelf and modifying it.

In most cases, the modification method rarely produces a package I'm willing to show off. Having the ability to create a custom package is something that 3D printing helps to deliver to any electronics developer. I am constantly amazed at what people are able to create with these things.

I finished off my box by painting the recessed letters with some white paint. It wasn't the best look, but it was definitely more visible. I could have printed in two colors with the FLASHFORGE Dreamer since it has dual extruders, but I'll save that for a future article/video. **NV**



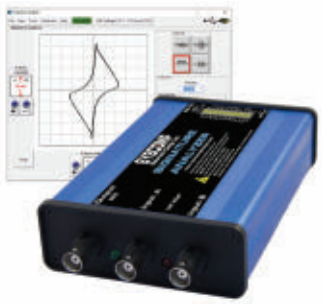
■ **FIGURE 10.** Assembled unit.

Saelig.com unique electronics

28 Years of Remarkable Products and Superior Customer Care

More Unique Products

Signature Analyzer



- Controlled via USB
- 5-in-1 Functions
- Includes Oscilloscope AWG, I/O
- WIN, MacOS, Linux
- Under $400

SIG-101

Oscilloscope, Waveform Generator, Digital I/O Port - All in One!

- Terrific educational tool with lessons/experiments
- AWG, I/O, PWM, FFT
- $99.95

CGM-101

SDS1102X Scope Free Serial Decode Option

- 1GSa/s Sampling
- 14 Mpts Memory
- FFT
- 8" LCD
- 60,000 wfm/s
- $424

Wide Selection of Spectrum Analyzers

- Desktop to USB stick versions cover up to 12GHz to find signal qualities or interference
- Starting at $518

Power Supplies - 30W to 1500W

- Desktop to rack-mount power supplies to fit almost any need.
- Programmable on/off voltage level cycles
- From $56

Automotive Diagnostic Equipment

- Picoscope test kits are designed to make diagnosing & testing vehicle electronics, components easy. Find faults fast!

EMC Precompliance Testing

- Check your PCB design before expensive formal testing to make sure you'll pass EMC specs with our affordable **solutions.**

PCB Repair Equipment

- Repair, don't throw away costly PCBs. BoardMaster systems are easy-to-use and can be used by anyone to find and repair PCB faults.

BY FRED EADY

Exploring the Nuances of Nordic Semiconductor's nRF52832's Timer and GPIO Peripherals

We have proven that the Nordic Semiconductor nRF52832 is relatively easy to activate in the role of a BLE (Bluetooth Low Energy) peripheral device. However, the most intriguing thing about the Nordic nRF52832 is its very powerful 32-bit ARM Cortex-M4F CPU, which is chunking bits at 64 MHz. This time around, instead of thinking radio, we'll focus on treating the nRF52832 as a very powerful microcontroller. Let's have some fun.

Flight-Tested Hardware

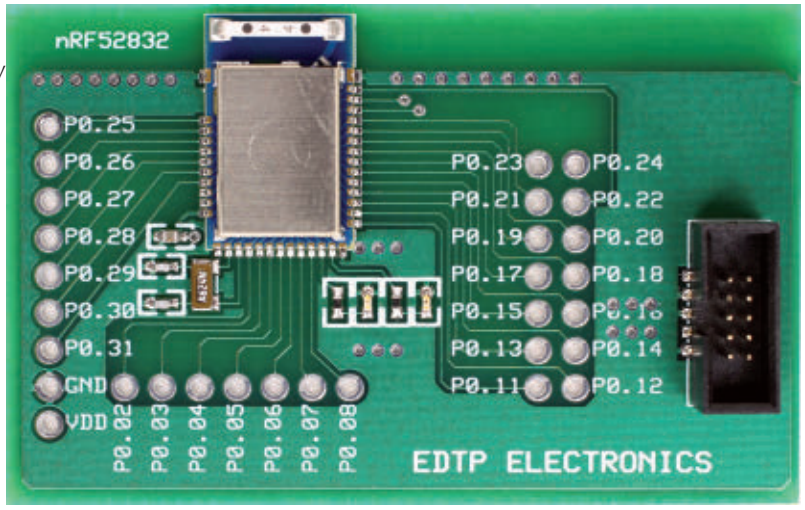
Since we've already designed and assembled some known working BLE hardware, why not use it as the target for the new GPIO firmware we're going to create. Our compiled ARM Cortex-M code will be loaded into the Raytac BLE module mounted on the nRF52832 development board shown in **Photo 1**. I've chosen to use the Raytac module because it is a true Nordic Semiconductor nRF52832 reference design housed in module form. Plus, Raytac was the only third-party module provider that readily offered the nRF52832 in module form.

The nRF52832 development board design includes a 32.768 kHz crystal and a pair of LEDs. The nRF52832 programming portal is composed of a single 1.27 mm 10-pin male header. The development board brings all available nRF52832 GPIO out to standard 0.1 inch pitch header positions. The vertical header position on the far left of the nRF52832 board can be used to mount it on a solderless breadboard.

Once mounted on a solderless breadboard, power for the nRF52832 can be obtained via the VDD and GND pins. The nRF52832 GPIO pins can be assigned to any of the peripherals. So, this vertical column of GPIO pins can be used to form a UART, SPI portal, I²C portal, PWM portal, ADC portal, or just plain GPIO. **Schematic 1** offers a full graphical depiction of the nRF52832 development board design.

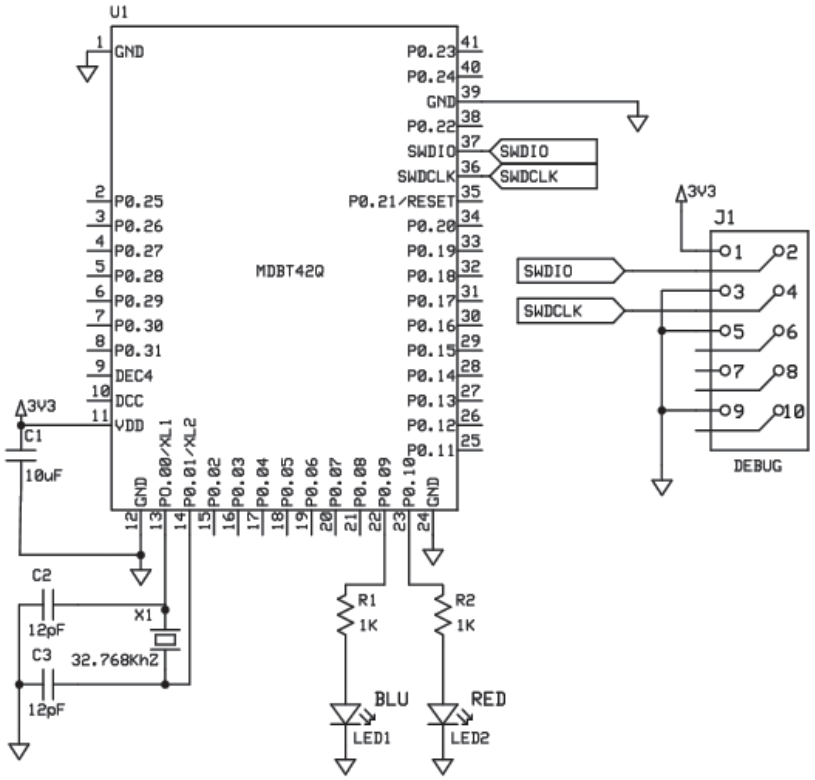
Firmware Development Platform

The Keil MDK for Cortex-M along with version 12 of the nRF5 SDK will be used to compose and compile our



■ **PHOTO 1.** The Raytac module contains a regulation Nordic nRF52832 supported by the appropriate capacitors, crystals, and inductors. An external 32.768 kHz crystal completes the hardware design.

Post comments on this article and find any associated files and/or downloads at **www.nutsvolts.com/magazine/article/December2016_DesignCycle**.

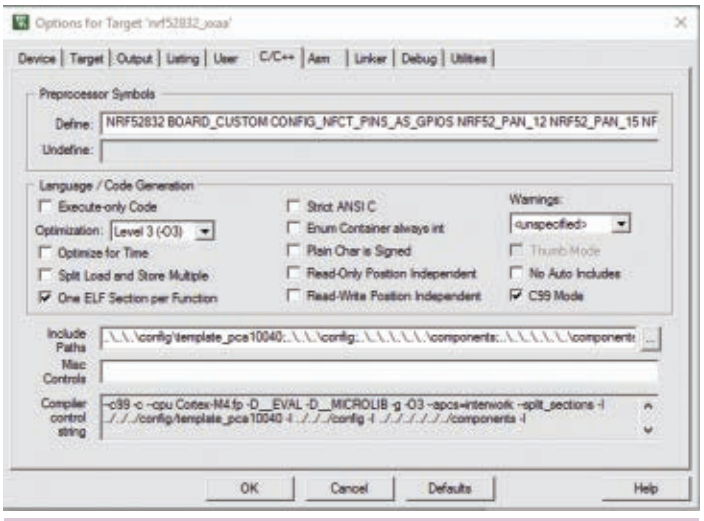


■ **SCHEMATIC 1.** This drawing is greatly simplified as all of the BLE radio circuitry is contained within the Raytac BLE module. An external 32.768 kHz crystal is included in the design to enable all of the nRF52832 clock features.

nRF52832 source code. The nRF5 SDK includes a wealth of examples. It also provides nRF5x drivers and nRF5x libraries. Installation of the Keil C compiler and Nordic SDK are detailed on the Keil and Nordic websites, so we won't spend any time on the firmware tool installation process.

We will base our initial code on a template that is found in the SDK *examples/peripheral* directory. Version 12 of the SDK introduced the *sdk_config.h* file and Configuration Wizard, which are used to configure the nRF drivers and libraries. This makes keeping up with who's on the BLE playing field a bit easier.

Programming and debugging duties will be handled by a Segger J-Link Pro. The J-Link Pro is fast, easy to use, and reliable. The Keil MDK fully supports the J-Link Pro.



■ **SCREENSHOT 1.** The Options window allows us to control how the hardware behaves. We can define things such as memory boundaries and what device will be used to program the nRF52832.

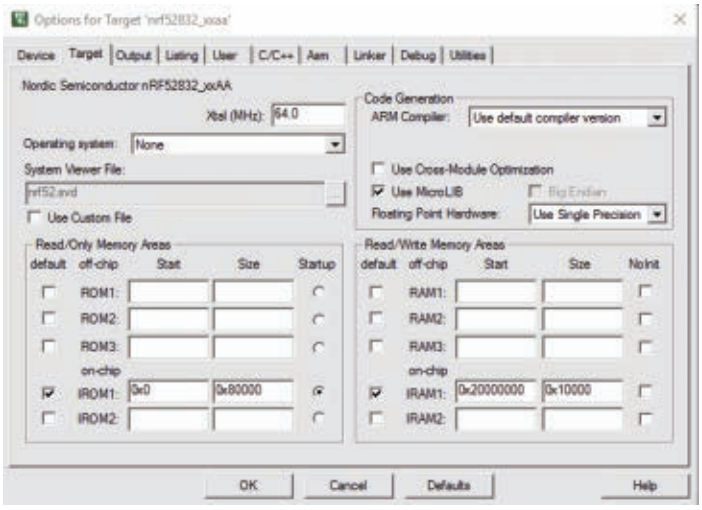
Setting Up the Compilation Options

The first order of business is to make sure we can use the nRF52832's GPIO pins assigned to drive the pair of LEDs native to the development board, which happen to be driven by GPIO pins 9 and 10. Out of the box, the nRF52832 is designed to drive an NFC device with the pins we have assigned to drive the blue and red LEDs. Since we have stepped out of the box with our LED assignment, we also need to register our LED I/O pins with the SDK's board support package firmware.

Disabling the NFC default pin assignment is as easy as typing *CONFIG_NFCT_PINS_AS_GPIOS* in the C/C++ options window. You can see that the NFC-disabling configuration phrase has been entered in the Preprocessor Symbols area of **Screenshot 1**. The board support package registration consumes two steps. The first step has been taken with the entry of *BOARD_CUSTOM* in the Preprocessor Symbols definition area, which can also be seen in **Screenshot 1**.

The second step in the board support package registration process entails creating a custom board support file called *custom_board.h*. The nRF5 SDK recognizes this file as the file in which our custom board peripheral pin assignments are kept. We have two LEDs. So, let's define them within our *custom_board.h* file:

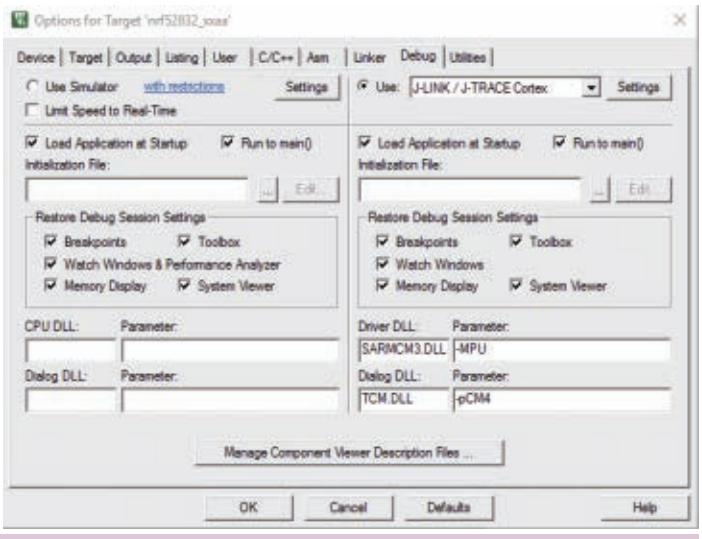
```
#define LEDS_NUMBER     2
#define LED_START       9
#define LED_1           9
#define LED_2           10
#define LED_STOP        10
#define BSP_LED_0       LED_1
#define BSP_LED_1       LED_2
```



■ **SCREENSHOT 2.** This window becomes very important when we load a SoftDevice. As you can see, with no SoftDevice loaded we currently have full run of the nRF52832's ROM and RAM areas.

If you take a look at **Schematic 1**, you will see that the nRF52832 development board's LEDs are attached to pins 9 and 10 of the Raytac BLE module. The SDK's board support package routines will manipulate the development board's LEDs using the BSP_LED_0 and BSP_LED_1 definitions. The board support package functions also depend on masks to control LED illumination. These masks are derived from our initial LED definitions:

```
#define BSP_LED_0_MASK (1<<BSP_LED_0)
#define BSP_LED_1_MASK (1<<BSP_LED_1)
```



■ **SCREENSHOT 3.** This is where we tell the Keil C compiler that we will be using a J-Link programmer/debugger.

```
#define LEDS_MASK    (BSP_LED_0_MASK | BSP_LED_1_
MASK)
// all LEDs are lit when GPIO is low
#define LEDS_INV_MASK  LEDS_MASK
```

Our *custom_board.h* file uses *boards.h* as its template. The *boards.h* file is part of the version 12 nRF5 SDK. The Nordic SDK examples contain enough information to allow us to pick and choose what we need to build our own custom files.

You will find that there are not many things related to the nRF52832 that you will have to write from scratch. A key thing to remember is that the Keil editor allows the programmer to immediately access the origins of any of the Nordic SDK types, functions, macros, structures, and definitions. The ability to instantly know what lies within a particular structure or function flattens the nRF52832 learning curve a bit.

The nRF52832 comes standard with 512 kB of Flash and 64 kB of RAM. This is verified in **Screenshot 2**. We will not require the services of a SoftDevice. So, we have the whole of the nRF52832 memory resources at our disposal.

Screenshot 3 completes our firmware tools setup. As I mentioned earlier, we will be using a J-Link Pro programmer/debugger to load and debug our nRF52832. If we missed anything, it's no problem to return to the Options windows and make it right.

Okay. Our hardware and firmware tools are primed and ready. So, let's take a stab at manipulating those development board LEDs:

```
nrf_gpio_pin_set(9);
```

That's all it takes to illuminate the nRF52832 development board's blue LED, which is driven by pin 9 of the nRF52832. Change the 9 to a 10 and the red LED will light up. To turn off the LEDs, use *nrf_gpio_pin_set(9)* and *nrf_gpio_pin_set(10)*.

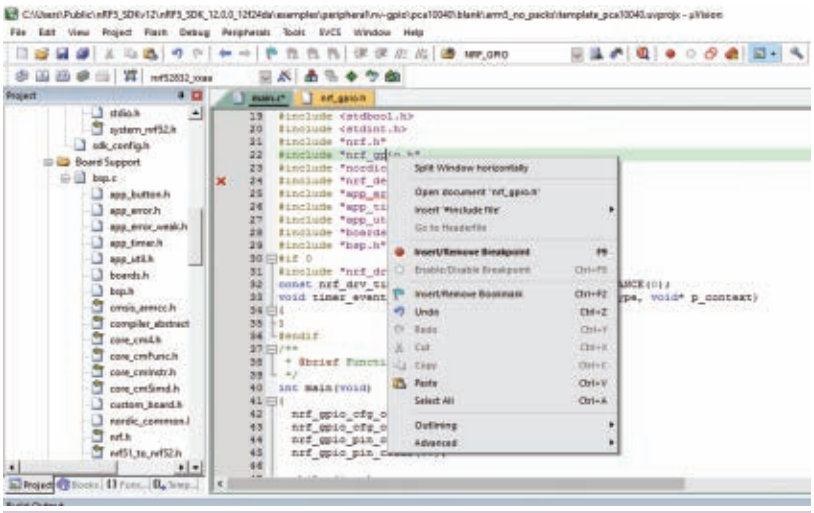
Before we can use the *pin set* and *pin clear* functions, we must configure pins 9 and 10 as output pins. Here is the code to turn both the blue and red LEDs on:

```
nrf_gpio_cfg_output(9);
nrf_gpio_cfg_output(10);
nrf_gpio_pin_set(9);
nrf_gpio_pin_set(10);
```

There are a couple of ways to discover how to control the nRF52832's GPIO pins. From the Keil text editor, you can click on the "nrf_gpio.h" portion of the #include "nrf_gpio.h" entry and open the *nrf_gpio.h* file. Within the file, you will find the prototypes for the functions we just used to configure and control the states of the I/O pins.

The second method is to go to the Nordic Semiconductor website and open up the nRF driver portion of the SDK documentation. The explanation of the code we just used to illuminate the LEDs can be found under GPIO Abstraction. I've captured the results of right-clicking on the "nrf_gpio.h" text in **Screenshot 4**.

The menu that appears allows many options, including opening the *nrf_gpio.h* file in the Keil text editor window. Recall that the human programmer can also get details of functions in the same way. In **Screenshot 5**, I have right-clicked on a function. Note that I can choose to be directed to the reference of the function or the initial definition of the function.



■ **SCREENSHOT 4.** This is the result of right-clicking on the "nrf_gpio.h" text in the Keil source editor. Note the *Open document "nrf_gpio.h"* menu item.

Hands-Off I/O

And I do mean hands-off. The nRF52832 is capable of manipulating its GPIO subsystem without the aid of its ARM CPU. This is done using the on-chip GPIOTE peripheral. The "TE" is short for Tasks/Events. The concept is relatively simple.

The output state of an nRF52832 GPIO pin can be changed upon the occurrence of a predefined event. Thus, the GPIO pin performs the "task" depending on the reception of an event. There are a number of possible events. Events can emanate from the radio, timers, other GPIO input pins, or other peripherals. Our event will trigger on a timer compare. With that, let's set up our timer:

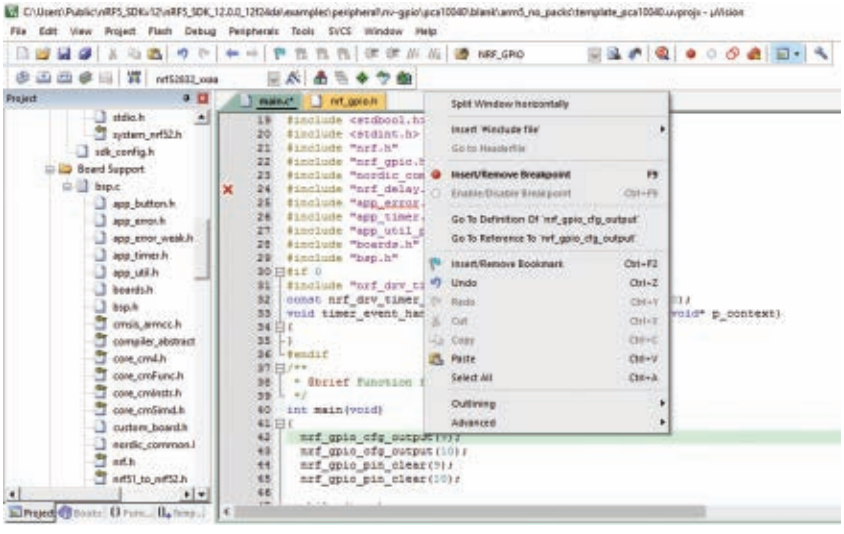
```
static nrf_drv_timer_t timer = NRF_DRV_TIMER_
INSTANCE(0);
void timer_dummy_handler(nrf_timer_event_t event_
type, void * p_context){}
```

We won't need the services of the timer handler. However, some of the timer related functions require the presence of a handler. The timer details are set up within the Configuration Wizard (*sdk_config.h*). Take a look at **Screenshots 6** and **6b**. Within both screenshots, the *Timer0* instance was enabled and the primary timer properties were set up. Our *Timer0* will operate with a 1 MHz clock in 32-bit timer mode.

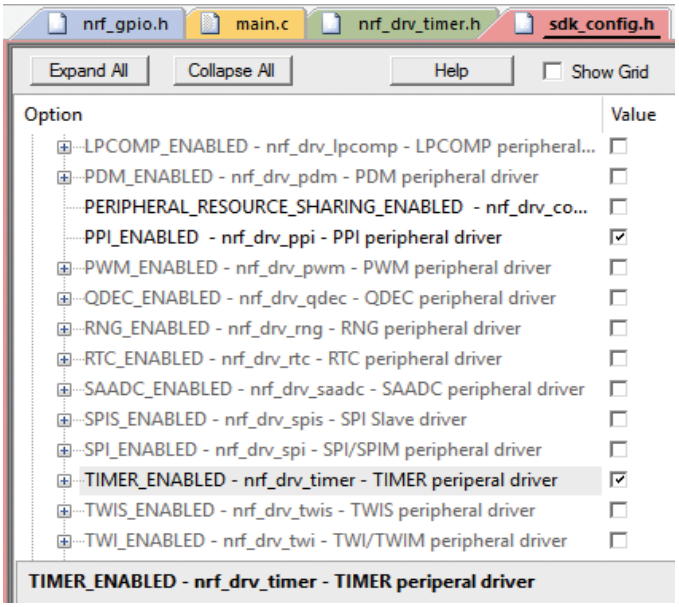
The path between the blue LED GPIO pin and the *Timer0* event is made using another ingenious nRF52832 feature. That "feature" is PPI (Programmable Peripheral Interconnect). To use PPI, we must supply an address for the task and an address for the event.

PPI uses the address information to form a link between the event and task. The event generates a signal that is routed to the task via the PPI connection. Here's the code that defines the task and event addresses and PPI channel:

```
static void initLED()
{
uint32_t timer_compare_evt_addr;
//event address variable
uint32_t gpiote_led_task_addr;
//task address variable
```



■ **SCREENSHOT 5.** Right-clicking on the text *nrf_gpio_cfg_output(9)* brings up the menu you see here. The programmer can click on *Go To Definition Of "nrf_gpio_cfg_output(9)"* and get the details of the function.

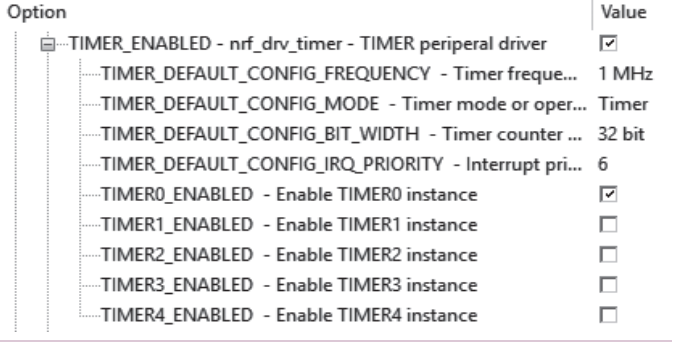


■ **SCREENSHOT 6.** The Configuration Wizard allows us to select the drivers and libraries we wish to include in our application. Here, we've turned on the PPI and Timer drivers.

```
nrf_ppi_channel_t ppi_channel;
//PPI channel type
ret_code_t err_code;
nrf_drv_gpiote_out_config_t config = GPIOTE_CONFIG_
OUT_TASK_TOGGLE(false);

err_code = nrf_drv_gpiote_out_init(blueLED9,
&config);
APP_ERROR_CHECK(err_code);
```

We have also specified how the GPIO pin driving the blue LED (pin 9) will behave. On the occurrence



■ **SCREENSHOT 6B.** The Timer driver needs some tweaking. As you can see, we have set the timer clock frequency to 1 MHz and set the timer resolution to 32 bits. Set at 6, this IRQ priority is next to the lowest priority setting of 7.

of an event, the blue LED will toggle. *GPIOTE_CONFIG_OUT_TASK_TOGGLE* is actually a macro:

```
#define GPIOTE_CONFIG_OUT_TASK_TOGGLE(init_high      \
 {                                                   \
  .init_state = init_high ? NRF_GPIOTE_INITIAL_      \
  VALUE_HIGH : NRF_GPIOTE_INITIAL_VALUE_LOW,         \
  .task_pin   = true,                                \
  .action  = NRF_GPIOTE_POLARITY_TOGGLE,             \
 }
```

The *GPIOTE_CONFIG_OUT_TASK_TOGGLE* macro argument (*init_high*) determines the initial state of the GPIO pin. A false argument sets the pin driving the blue LED initially low, which extinguishes the LED. The macro contents also identify the GPIO pin driving the blue LED as a task-controlled GPIO pin that will toggle on the reception of an event.

The contents of the *GPIOTE_CONFIG_OUT_TASK_TOGGLE* macro are used to populate the *nrf_drv_gpiote_out_config_t* structure:

```
typedef struct
{
  nrf_gpiote_polarity_t action;
/**< Configuration of the pin task. */
  nrf_gpiote_outinit_t  init_state;
/**< Initial state of the output pin. */
  bool     task_pin;
/**< True if the pin is controlled by  */
/**< a GPIOTE task.                    */
} nrf_drv_gpiote_out_config_t;
```

Let's turn our attention back to *Timer0*. We want to make sure we blink the blue LED slow enough for the human eye to pick it up:

```
nrf_drv_timer_extended_compare(&timer, (nrf_
timer_cc_channel_t)0, 500 * 1000UL, NRF_TIMER_
SHORT_COMPARE0_CLEAR_MASK, false);
```

Each tick of the 1 MHz clock driving *Timer0* is 1 µS long. So, to blink the blue LED every half second, we need 500,000 ticks.

NRF_TIMER_SHORT_COMPARE0_CLEAR_MASK is a macro representing a shortcut. A shortcut is simply a direct path that is established between a task and event. In this case, every time *Timer0* triggers a compare event, a shortcut path is used to automatically clear the timer. The Boolean false at the end of the function call disables the generation of an interrupt on a *Timer0* compare event.

Recall that this whole PPI/GPIOTE process is based on task and event addresses being logically connected. We have already laid the groundwork. Now, let's build those

connections. First, let's bring up one of the 20 user-defined PPI channels available to us:

```
err_code = nrf_drv_ppi_channel_alloc(&ppi_
channel);
APP_ERROR_CHECK(err_code);
```

The next step of building the PPI/GPIOTE infrastructure requires us to retrieve the event and task addresses:

```
timer_compare_evt_addr = nrf_drv_timer_event_
address_get(&timer, NRF_TIMER_EVENT_COMPARE0);
gpiote_led_task_addr = nrf_drv_gpiote_out_task_
addr_get(blueLED9);
```

Now that we have allocated an unused PPI channel and retrieved the event and task addresses, we can put the PPI channel and task/event addresses together logically:

```
err_code = nrf_drv_ppi_channel_assign(ppi_
channel, timer_compare_evt_addr, gpiote_led_task_
addr);
APP_ERROR_CHECK(err_code);
```

The *APP_ERROR_CHECK* function verifies that our function request was valid and processed without error. It is good programming practice to include this check when coding nRF52832 applications.

Everything is in place. All that's left to do is flip the switch and turn it all on:

```
err_code = nrf_drv_ppi_channel_enable(ppi_
channel);
APP_ERROR_CHECK(err_code);

nrf_drv_gpiote_out_task_enable(blueLED9);
} //end of initLED function
```

The *initLED* function will be called from the *main* function. For the lights to come on, we will need to throw some additional switches in the *main* function before we call the *initLED* function:

```
  ret_code_t err_code;

  err_code = nrf_drv_ppi_init();
  APP_ERROR_CHECK(err_code);

  err_code = nrf_drv_gpiote_init();
  APP_ERROR_CHECK(err_code);

  nrf_drv_timer_config_t timer_cfg = NRF_DRV_
  TIMER_DEFAULT_CONFIG;
  err_code = nrf_drv_timer_init(&timer, &timer_
  cfg, timer_dummy_handler);
  APP_ERROR_CHECK(err_code);

    initLED();

    nrf_drv_timer_enable(&timer);
```

As you can see, we initialize the major players at the beginning of the *main* function. Recall that we preconfigured *Timer0* with the Configuration Wizard. The *NRF_DRV_TIMER_DEFAULT_CONFIG* macro pulls from the *Timer0* entries we made in the Configuration Wizard, which were deposited in *sdk_config.h*.

Note that we have been populating structures (sometimes using macros) and passing the contents of the structures to functions. The beauty to this is that all you have to do is right-click on a structure you want to study more closely within the Keil text editor and walk through the structure contents from the initial structure declaration. Here's the remainder of the *main* function code:

```
while (true)
{
   // Do nothing.
}
```

I loaded the compiled code and the blue LED is blinking.

More to Come

In the next installment of Design Cycle, we will continue our exploration of the nRF52832. **NV**

Keil Cortex-M MDK
Keil
www.keil.com

Raytac MDBT42Q
nRF52832 Module
Raytac
www.raytac.com

J-Link PRO
Segger
www.segger.com

Continued from page 7

many switches upon release.

I wired up your choice of debounce components and attached a digital storage oscilloscope. I selected a quality switch that gives me readily repeatable contact bounce characteristics that happen to be around two milliseconds in duration. Predictably, on press, the hash was effectively quelled, but upon release the scope trace showed much contact bounce still present, which would appear as multiple button presses to the microcontroller's input pin. May I ask why you chose to use the diodes in your debounce circuit? Simply removing the diodes would make a substantial improvement in bounce handling.

Using the discussion, its math equations, and experimentation, I came up with component values that I would utilize for switch debouncing in a future project. Personally, I really don't like the idea of the bypass diode. I would use a capacitor of 0.47 µF, a discharge resistor of 18.2K, and a charge resistor of 2.74K with no diode. This gives contact bounce protection for 14 ms on the press and for 16 ms on the release. In your opinion, do you think this would be overkill if used in your Numitron clock?

Thank you for a wonderful article, Bill!

Judy May W1ORO
Union, KY

Thank you, Judy! I don't know how to respond to such praise! Following are my answers to your questions in order.

Indeed! Sad is the day you don't learn something new. Even in "failure," something is to be learned as Edison purportedly once proclaimed: "I have not failed 700 times. I've succeeded in proving 700 ways how not to build a light bulb." Words I live by since most of my attempts at achieving any goal are usually preceded by many discoveries about how it does *not* work!

The Weller WESD51 is a wonderful soldering unit; excellent quality, light, fast heating, and reasonably priced. Highly recommended for anyone looking to buy a set. The ETH tip is a screwdriver (or also referred to as a chisel) point tip; the one I use (and prefer) for SMD work is a long conical type (ETS) and a shorter conical one (ETU). Combined with a fine solder (I use 0.020"), soldering the parts in my Numitron project is rather doable with a bit of practice (and using a head band type magnifying lens).

Ten seconds per year is an almost astonishing accuracy for that type of design. With all the stuff my PIC has to do in controlling the LED display and the Numitron display, I was not able to keep the time interrupt in check when using the main crystal as the time base. Using the 32.768 kHz crystal allowed me to relax the timing constraints in the main program somewhat (the PIC does not support priority interrupts). Very possible a better programmer than I could do it; I had the parts and this seemed like a good solution at the time. Future clock designs I'm working on will utilize a Maxim DS3231 I^2C dedicated calendar/clock chip with built-in TCO and a battery

THE OWNERSHIP, MANAGEMENT, AND CIRCULATION STATEMENT OF NUTS & VOLTS MAGAZINE, Publication Number: 1528-9885 is published monthly. Subscription price is $26.95. 7. The complete mailing address of known office of Publication is T&L Publications, Inc., 430 Princeland Ct., Corona, Riverside County, CA 92879-1300. Contact Person: Larry Lemieux. Telephone: (951) 371-8497. 8. Complete Mailing address of Headquarters or General Business Office of Publisher is T&L Publications, Inc., 430 Princeland Ct, Corona, CA 92879. 9. The names and addresses of the Publisher, and Associate Publisher are: Publisher, Larry Lemieux, 430 Princeland Ct., Corona, CA. 92879; Associate Publisher, Robin Lemieux, 430 Princeland Ct., Corona, CA 92879. 10. The names and addresses of stockholders holding one percent or more of the total amount of stock are: John Lemieux, 430 Princeland Ct., Corona, CA 92879; Lawrence Lemieux, 430 Princeland Ct., Corona, CA 92879; Audrey Lemieux, 430 Princeland Ct., Corona, CA 92879. 11. Known Bondholders, Mortgagees, and other security holders: None. 12. Tax Status: Has not changed during preceding 12 months. 13. Publication Title: Nuts and Volts 14. Issue Date for Circulation Data: October 2015-September 2016. 15. The average number of copies of each issue during the proceeding twelve months is: A) Total number of copies printed (net press run); 30825 B) Paid/Requested Circulation (1) Mailed Outside County subscriptions: 19965 (2) Mailed In-County subscriptions: 0 (3) Paid Distribution Outside the Mail including Sales through dealers and carriers, street vendor, and counter sales and other paid distribution outside USPS: 5631 (4) Paid Distribution by other classes of mail through the USPS: 0; C) Total Paid Distribution: 25596; D) Free or Nominal Rate Distribution by mail and outside the mail (1) Free or Nominal Rate Outside-County Copies: 0 (2) Free or Nominal Rate In-County Copies: 0 (3) Free or Nominal Rate Copies Mailed at other classes through the USPS: 0 (4) Free or Nominal Rate Distribution Outside the mail: 1,321; E) Total Free or Nominal Rate Distribution: 1,321; F) Total Distribution: 26917; G) Copies not distributed: 3908 H) Total: 30825; Percent paid circulation: 95.09%. Actual number of copies of the single issue published nearest the filing date is September 2016; A) Total number of copies printed (net press run) 30508; B) Paid/Requested Circulation (1) Mailed Outside County subscriptions: 19382 (2) Mailed In-County subscriptions: 0 (3) Paid Distribution Outside the Mail including Sales through dealers and carriers, street vendor, and counter sales and other paid distribution outside USPS: 6831 (4) Paid Distribution by other classes of mail through the USPS: 0; C) Total Paid Distribution: 26213; D) Free or Nominal Rate Distribution by mail and outside the mail (1) Free or Nominal Rate Outside-County Copies: 0 (2) Free or Nominal Rate In-County Copies: 0 (3) Free or Nominal Rate Copies Mailed at other classes through the USPS: 0 (4) Free or Nominal Rate Distribution Outside the mail: 800; E) Total Free or Nominal Rate Distribution: 800; F) Total Distribution: 27013; G) Copies not distributed: 3495; H) Total: 30508; Percent paid circulation: 97.04%. I certify that these statements are correct and complete. Lawrence Lemieux, Publisher - 10/31/2016.

CLASSIFIEDS

AUDIO/VIDEO

PARTS EXPRESS
YOUR ELECTRONICS CONNECTION
OVER 18,000 ELECTRONIC PARTS IN STOCK
Speakers
Components
Project Accessories
Tools and Tech Aids
Save $5 on $75 order at parts-express.com!
Promo code: NVOCT16
*Offer expires 12/31/16
parts-express.com
1-800-338-0531

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!
Digital Equipment Corp. and compatibles.
Buy - Sell - Trade
Call Keyways 937-847-2300
or **email buyer@keyways.com**

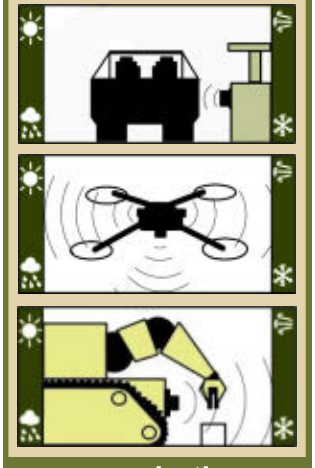
Subscribe today!
www.nutsvolts.com

APPS/BOOKS

Build your own Monitoring and Alarm system. The Android app Monitoring reads hard wired contact statuses and then transmits user defined alarm messages. The messages are sent by Gmail to your phone or to a computer. The app requires Android M or higher, an OTC cable, and an external MIDI, ASCII, or development board. Search for "Monitoring and Znidarsic" to view the app at Amazon and Google. The book "Electronics Project" is also available at "Znidarsic Science Books" at Amazon. Post your experience on the Amazon discussion.

ROBOTICS

Need sensors?



www.maxbotix.com

LIGHTING

backup connection though. As they say, there are many ways to skin a cat.

The author used a 74AHCT14 hex Schmitt trigger — an interesting choice since that is an automotive version of the standard HCT CMOS family. Not knowing the manufacturer of his device, I just picked one (NXP Semiconductors), and it seems that for that device at Vcc 4.5V, the low transition is around 0.44V and the high transition around 3.8V. These numbers seem to vary a bit depending on the output current. Either way, it seems he did indeed mix up the high/low transition point. I did not go as far as recalculating his math to see if he also mixed it up there as well. Later on PG14 when talking about the 7414, he has it correct. Good catch though.

At my age, it is becoming a chore remembering what I did yesterday, let alone recall my thought process over a year ago :-). The quality of a switch has a lot to do with the bounce period; also, there is usually less bounce when a switch opens vs. closes. I did a fair bit of web research looking for examples of what others did.

Good point. An inferior switch could possibly cause problems, but since the board is designed for a small (and usually pretty fast) tactile switch, I don't think many will attempt to use something else. Your point here is valid though too.

The software is actually such that the bounce on release is not much of an issue, but I see no downside in leaving the diode out. Keep in mind that this clock is the product of an amateur (me), not the product of an accomplished engineer. The important part here is that it works, and was fun to build.

With respect to this Numitron clock project, I would say changes are not required; it would be akin to fixing something that works. With respect to using this design in other or future projects, the design (and most definitely chosen values) must be re-evaluated. For that reason, the web link was included so that anyone wishing to implement a similar design in their own project would have the tools and information to build their own version of the circuit rather than copying my design. I could have easily relied on software debounce only, but at some point I decided this would be fun to do.

On a similar vein, nobody questioned why I used a counter to sequence the LEDs, and a shift register to sequence the Numitron display. Either solution would have worked for both, but for no other reason than my own satisfaction I used both methods in my design. Such is the freedom in amateur projects vs. a professional project. The fact that it made you think and question my design shows the value of sharing working but imperfect hobby designs.

I am glad you enjoyed the article. I have had a lot of entertaining hours designing and building it. I too learned a lot during its development, and it gave me confidence in tackling my next project. Being able to share it with others is icing on the cake.

They also make nice presents to friends and family (which, once given away, allows me to build another one since my SO will tolerate only so much in the house). Many of my projects are difficult to share in a magazine, either because they are built with one-off components, or simply have too many bugs. This was one of my projects that I felt was ready for sharing.

Bill van Dijk

The Numitron is now available as a kit from the *Nuts & Volts* Webstore!

The kit includes:
- PCB
- Programmed PIC16F876A-I/SO Chip
- 12V 1.5A Power Supply
- Six IV-9 IV-9-Numitron Tubes
- Sixty LEDs plus 10 extra LEDs
- Over 65 SMT and through hole components.

Buy now for only $119.95!

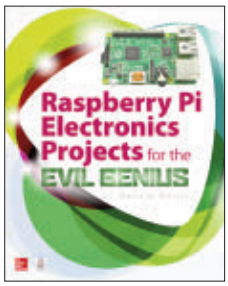
Go to: http://store.nutsvolts.com/project-kits/sku17430

For complete product details, visit us online!!

The Nuts & Volts WEBSTORE

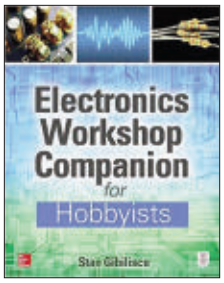
GREAT FOR DIYers!

Raspberry Pi Electronics Projects for the Evil Genius
by Donald Norris



This fully illustrated TAB guide shows how to construct and program all kinds of fun and innovative gadgets with your Raspberry Pi. *Raspberry Pi Electronics Projects for the Evil Genius* features 10 complete projects that showcase cool RasPi applications in computing, communications, robotics, photography, and video. Each inexpensive project includes a detailed list of materials, sources for parts, schematics, and clear step-by-step assembly and programming instructions.
$25.00

Electronics Workshop Companion for Hobbyists
by Stan Gibilisco

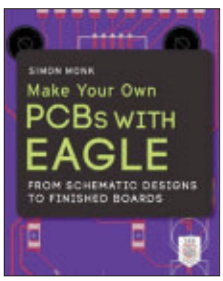


In this practical guide, electronics expert Stan Gibilisco shows you, step by step, how to set up a home workshop so you can invent, design, build, test, and repair electronic circuits and gadgets. *Electronics Workshop Companion for Hobbyists* provides tips for constructing your workbench and stocking it with the tools, components, and test equipment you'll need. Clear illustrations and interesting do-it-yourself experiments are included throughout this hands-on resource.
$25.00

Arduino: A Beginner's Guide To Programming Electronics
by Chuck Hellebuyck

Arduino is one of the most popular microcontroller development boards available. It's more than just a module though, as the heart of Arduino is the software you run on your computer and use to create sketches (or programs) in the C programming language, with many pre-built functions. Chuck explains all this in very simple terms to help you get started programming your own electronic applications. His down-to-earth style of explaining technology makes it a great place to get started with programming electronics. **$15.00**

Make Your Own PCBs with EAGLE
by Eric Kleinert



Featuring detailed illustrations and step-by-step instructions, *Make Your Own PCBs with EAGLE* leads you through the process of designing a schematic and transforming it into a PCB layout. You'll then move on to fabrication via the generation of standard Gerber files for submission to a PCB manufacturing service. This practical guide offers an accessible, logical way to learn EAGLE and start producing PCBs as quickly as possible.

$30.00

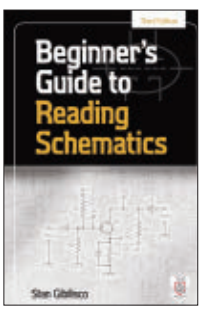
Programming the Intel Edison: Getting Started with Processing and Python
by Donald Norris

Learn To Easily Create Robotic, IoT, and Wearable Electronic Gadgets!

Discover how to set up components, connect your PC or Mac, build Python applications, and use USB, WiFi, and Bluetooth connections. Start-to-finish example projects include a motor controller, home temperature system, robotic car, and wearable hospital alert sensor.

$20.00

Beginner's Guide to Reading Schematics, 3E
by Stan Gibilisco

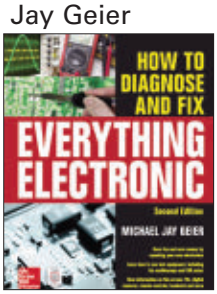


Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.

$25.00

How to Diagnose and Fix Everything Electronic
by Michael Jay Geier

A Fully Revised Guide to Electronics

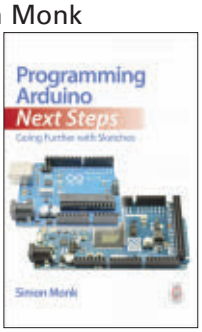


Repair all kinds of electrical products, from modern digital gadgets to analog antiques, with help from this updated book. The Second Edition offers expert insights, case studies, and step-by-step instruction from a lifelong electronics guru. Discover how to assemble your workbench, use the latest test equipment, zero in on and replace dead components, and handle reassembly. **$24.95**

Programming PICs in Basic
by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller. **$14.95**

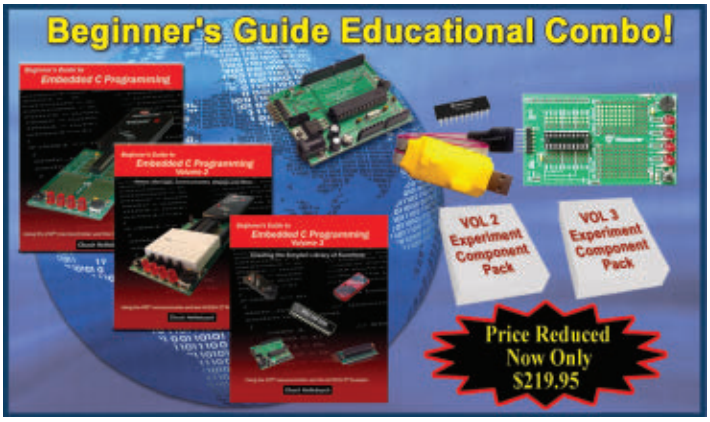
Programming Arduino Next Steps: Going Further with Sketches
by Simon Monk



In this practical guide, electronics guru Simon Monk takes you under the hood of Arduino and reveals professional programming secrets. Also shows you how to use interrupts, manage memory, program for the Internet, maximize serial communications, perform digital signal processing, and much more. All of the 75+ example sketches featured in the book are available for download. **$20.00**

Order online @ **store.nutsvolts.com**
Or CALL **1-800-783-4624** today!

EDUCATIONAL



CD-ROM SPECIAL

All New
TECHFORUM
Online At
www.nutsvolts.com/tech-forum

>>> QUESTIONS

Fridge Alert

My young son frequently leaves the fridge door open. I'm an electronics beginner and I'd like to build a simple circuit to alert me when that happens. It should be simple enough that I could build it with the help of my son, so he learns two lessons in one sitting. Schematic and design appreciated!

#12161 **Michael Calhoun**
Odessa, MO

Speaker Sharing

Does anyone have a simple circuit that will allow me to mix my iPod audio out with my computer's audio to play through the same speakers? I want to avoid un-plugging/re-plugging just to hear some music.

#12162 **Jonnie Vanalstyne**
Manchester, NH

Hit By A Speeding Card

My Raspberry Pi Model III uses a microSD card. The cards are available in different "speeds." Is there any significance to how "fast" a card I get?

#12163 **Rick Holtz**
Grand Rapids, MI

Getting Started With SDR

I want to start experimenting with Software Defined Radio (SDR) There seems to be a fair amount of info, but I'm still not sure where to start. What are the minimum requirements to get started on a small budget that the wife will live with? Is there a best "starter kit?"

#12164 **Amaranto Melgar**
Wetmore, KS

>>> ANSWERS

[#8166 - August 2016]
Water Wise

I would like to add a "rain detector" to my sprinkler system that would automatically disable the sprinklers. The system uses an old-style mechanical "dumb" timer. How about a simple circuit to do the job?

You can buy a normally closed SPST switch containing a water absorbing material that expands when wet to open a microswitch. This is placed in between the common side of the valves and the return line. The switch can be adjusted to open after an appropriate amount of rain has fallen. They are sold by most sprinkler manufacturers in home improvement and hardware stores.

Horton Prather
Buford, GA

[#8167 - August 2016]
Camptronics

It has fallen to me to teach a short course on electronics for a summer camp program near our home. I would welcome suggestions for curriculum! The kids are ages 8-12.

#1 Suggestion depending on length of camp:

1. Build an LED flashlight.
2. LED flashlight with dimmer.
3. Crystal radio.
4. Crystal radio with one-transistor amplifier.

Chip Veres
Miami, FL

#2 Summer is about over so this might be useful for next year, but this is what I would teach kids if I had about a week to do it.

DC/AC:
- Batteries and LEDs
- Types of electricity
- DC, AC line power
- Power line safety
- Tesla coil RF

Series and Parallel Game:
- Lines on floor
- Switches
- Rules
- Leader

AC Line Safety:
- Overloads
- Wet areas
- Reducing power use with CFL and LED bulbs

Takeapart Day:
- VCRs
- DVD players
- Computers and hard drives
- Problems with TVs — High voltage breaking glass
- Dangers of microwave ovens and how to avoid them with proper supervision

The series and parallel game is played on a gym floor or grass area with lines laid out with painter's tape or yarn and 4" nails. Use small rugs for batteries and lamps. Have kids to man the switches which are yarn sections with weights on the free ends.

The teacher yells out "switch one close," etc., then the switches move and the kids stationed along the "wires" move like electrons if the circuit is complete. Do a short circuit too where the kids have to move quickly and the teacher breaks the circuit and yells stop/sit down. This comes last, followed by a comment and question time. This is modeled after the on-stage demos that are part of Harvard's famous CS 50 class.

Have a local fire official do the line voltage safety lesson because they do this all the time.

Be sure to have tiny torx drivers for taking the super magnets out of hard drives.

I would like to see some guidelines for the Takeapart Day like:
- Unplug all wires rather than cut them
- Watch out for anything that has ink in it or on it, and wear gloves while removing that part to the

Send all questions and answers by email to **forum@nutsvolts.com** or via the online form at **www.nutsvolts.com/tech-forum**

trash can
- Recycle everything not saved for parts.

I hope that you find this useful.

Dale Freye
via email

#3 Recently, I too was asked to make an electronics presentation for the age group you mentioned. The library staff person referred to me as a docent, or lay presenter. The areas I get excited about would seem not to transfer well with this age group, so I needed help. Thankfully, I was well advised by the youth librarian who assisted me in creating and talking about very basic electrical experiments that she hoped would conclude with a hands-on soldering demonstration.

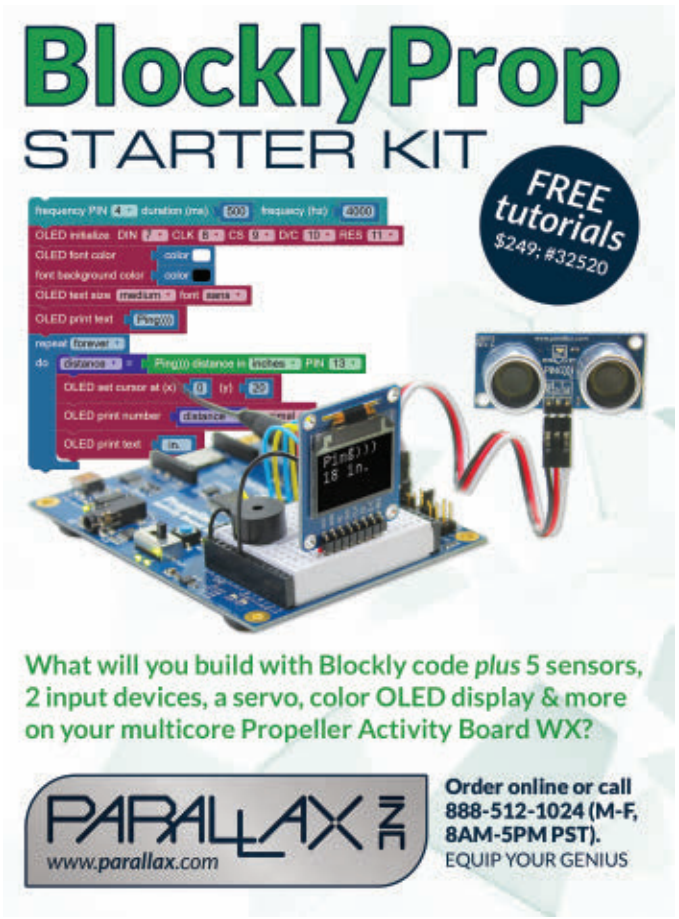
The first experiment employed a basic compass with five or so hand wound turns of 30 ga magnet wire circling it, powered with one AA battery. So, why does this make the needle spin? Discussion about magnetism and electrical relationships. Can anyone envision a motor in this (hold just one or two seconds to keep wire heating reduced)?

The next presentation offered the very simplest circuit with a 'schematic' drawing representing the actual physical setup of a power source, wire connections, and a (lamp) load. Then, a switch is added and drawn. Some discussion then follows about how current behaves and flows in this circuit.

The third presentation involved a zero center galvanometer (±.001A). Those large clear classroom kind work great. The terminals connected to the winding of a nail or screw type electromagnet with a small magnet fluxing a field to induct current, then swinging the needle both directions toward a basic conception of Alternating Current. Good time to mention diode rectification in the ensuing discussion.

The last was a very well-received LED yard light demonstration: a small solar panel with an electronic control circuit connected to the zero-center meter showing current direction, both powering a white LED and charging a battery configured like **Figure 1.**

The kids responded well using objects and their hands to shade the photovoltaic panel, watching the



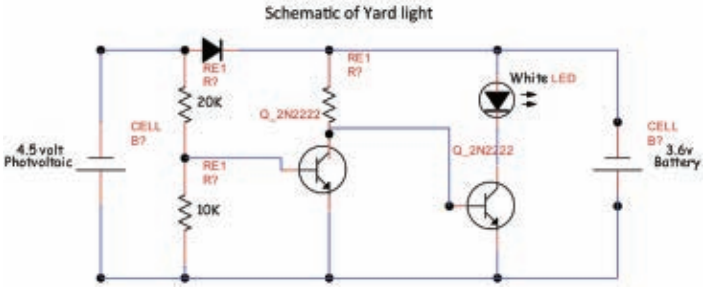


FIGURE 1. Just to the right of the white LED junction, between the 3.6 volt battery, insert the zero-center galvanometer. A 3-4 ohm shunt between the ± meter connections offered dramatic visual current direction. Raise or lower shunt resistance as needed.

meter swing and the white LED turn on and off; either charging or emitting light with only a simple transistor circuit. As they were marveling, I offered a brief description of the transistor operation in a schematic drawing describing the details of on-off (Schmitt) switching.

We DID have time to solder at the end. With a circuit board clamp at a comfortable angle, I used a 15 watt Weller with the LED lights, put my glasses on describing the need for a clean, tinned, tip and began soldering a through-hole PCB describing the process as I soldered each tiny pad. Would you like to try this? They all did, and I get teary in how excited they were when they completed their section of the many other pads. At the end of our class, the youth librarian shook my arm off in thanks, but I easily enjoyed it as much. It will be easier than you think. Have fun.

Michael Greenlee
via email

[#9161 - September 2016]
I Got The X10 Blues

My X-10 home automation gear has been reliable for years but just recently has become intermittent. Is there a way to test for what might be interfering with the operation?

#1 X-10 was developed back in the '70s as the first consumer power line carrier (PLC) home automation product. While newer and more robust systems are available, X10 remains popular due to its simplicity and low cost. Back in the '70s, we didn't have the plethora of wall warts, PC/printer/monitor power supplies, or CFL and LED bulbs — all of which tend to either absorb X10 signals or put noise on the power line or both!

All of these signal suckers and/or noise generators can swamp the X10 signal resulting in the problems you're experiencing. There are filters, amplifier/repeaters, and couplers to resolve these issues, and I'm happy to report that after over 30 years of using X10, I've been able to maintain reliable operation even in fairly large homes.

Before adding any of the above devices, the trick is to identify the source(s) of the problem. At the time the intermittent problem began, think about what changed in the house. A new TV, computer, monitor, or phone/tablet charger? Conversion to CFL or LED bulbs? If any of those get a yes, try removing them and testing. Plug-in filters will help.

Does the problem only happen at certain times? For example, when my in-ground incandescent pool light burnt out, I replaced it with an LED bulb. When it was on, it generated sufficient noise to render X10 useless to a number of locations. The solution was to add a hash choke in series between the switched wire from the X10 module and the light. Problem solved.

Another very important consideration is coupling from one side of the power line to the other. As we all know, residential service consists of two 120 volt legs (L1 and L2) from the street transformer to the electrical panel where roughly half the circuits are served by each leg. Electric ovens, water heaters, and central A/C units are connected to both legs to provide 240 volt service. If an X10 controller is plugged into an outlet on L1 and the lamp module is plugged into an outlet on L2, the signal needs to find a bridge to get from one leg to the other. That bridge can be a 240 volt appliance if it's on, or the street transformer itself — a long run that attenuates the signal.

The solution is to use a coupler at the electrical panel to provide a reliable path for the signal to reach every circuit. The very best solution that will overcome a multitude of problems is the installation of an XTB-IIR which is both a coupler and X-10 signal amplifier. It's available both as a kit or fully assembled from **www.jvde.us**. Building the kit is within the capabilities of most *Nuts & Volts* readers. That site offers a number of solutions and some great tutorials on X10 troubleshooting. I've been using both their site as a resource and their products for many years with great success.

Bruce Robin
Naples, FL

#2 I was having the same problem, with flakier and flakier performance. Some online research indicates that the problem is the result of the ever-increasing number of AC power line operated digital devices we use — each of which can feed noise back into the power line, potentially causing problems for other devices.

To prevent this, AC power line connections use a lot of filtering components (inductors and bypass capacitors) to prevent the digital noise from traveling back into the AC power line, but the bypass capacitors essentially short circuit the 120 kHz X-10 signals so they can't reach the devices they are intended to control. It's time to switch to Z-wave or one of the other radio-based control systems.

Daniel Duncan
San Gabriel, CA

ELECTRONET

Wanna read even more about electronics? Check out our book selection at
store.nutsvolts.com

USB **Add USB to your next project-- It's easier than you might think!** DLP Design
USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com

For the ElectroNet online, go to www.nutsvolts.com click *Electro-Net*

ANSWERS TO NAME THAT PART on page 38:

1b, 2b, 3a, 4c, 5b, 6c, 7a, 8a, 9c, 10c, 11c, 12b, 13c, 14b, 15a, 16a, 17a, 18b

ADvertiser INDEX

■LOOK FOR ■SEARCH FOR ■FIND **Find your favorite advertisers here!**

BATTERIES/CHARGERS
Hitec 2

BUYING ELECTRONIC SURPLUS
All Electronics Corp. 19

CIRCUIT BOARDS
Accutrace 66
AP Circuits 10
ExpressPCB 7
Front Panel Express LLC 10
Saelig Co., Inc. 51

COMPONENTS
All Electronics Corp. 19
Chaney Electronics 39
NTE Parts Direct 59
Saelig Co., Inc. 51
ServoCity 67

DATA LOGGING
Measurement Computing 15

DESIGN/ENGINEERING/ REPAIR SERVICES
Accutrace 66
AP Circuits 10
Cleveland Institute of Electronics .28
ExpressPCB 7
Front Panel Express LLC 10

DEVELOPMENT PLATFORMS/TOOLS
Anaren .....Back Cover
Technologic Systems 28

EDUCATION
Cleveland Institute of Electronics .28
Command Productions 33
Parallax 63
PoLabs 19

EMBEDDED SYSTEMS
Measurement Computing 15
Saelig Co. Inc. 51
Technologic Systems 28

ENCLOSURES
Front Panel Express LLC 10

LCDs/DISPLAYS
EarthLCD 21
Saelig Co., Inc. 51

MICROCONTROLLERS / I/O BOARDS
Technologic Systems 28

MISC./SURPLUS
All Electronics Corp. 19

MOTORS / MOTOR CONTROL
Hitec 2
ServoCity 67

RF TRANSMITTERS/ RECEIVERS
Lemos International 11

ROBOTICS
Chaney Electronics 39
Cleveland Institute of Electronics .28
Hitec 2
Parallax 63
ServoCity 67

TEST EQUIPMENT
Chaney Electronics 39
NTE Parts Direct 59
Pico Technology 3
PoLabs 19
Saelig Co., Inc. 51

TOOLS
Bench Werx 63
PanaVise 11
PoLabs 19

WIRE, CABLE AND CONNECTORS
All Electronics Corp. 19

WIRELESS PRODUCTS
Anaren .....Back Cover
Lemos International 11
Technologic Systems 28

Accutrace 66
All Electronics Corp. 19
Anaren .....Back Cover
AP Circuits 10
Bench Werx 63
Chaney Electronics 39
Cleveland Institute of Electronics 28
Command Productions 33
EarthLCD 21
ExpressPCB 7
Front Panel Express LLC 10
Hitec 2
Lemos International 11
Measurement Computing 15
NTE Parts Direct 59
PanaVise 11
Parallax 63
Pico Technology 3
PoLabs 19
Saelig Co., Inc. 51
ServoCity 67
Technologic Systems 28

PRINTED CIRCUIT BOARDS

THINK YOU CAN FIND PCB PRICES THAT BEAT OURS?

WE DARE YOU.

THERE ARE NO GAMES INVOLVED IN OUR PRICING

Our Capabilities:

- From same day quick turn prototype to production in under 10 days
- Full CAD and CAM review plus design rule check on ALL Gerber files
- Materials: Fr4, Rigid, Flex, Metal Core (Aluminum), Polymide, Rogers, Isola, etc.
- HDI Capabilities: Blind/Buried Mocrovias, 10+N+10, Via-in-Pad Technology, Sequential Lamination, Any Layer, etc.
- Our HDI Advantage: Direct Laser Drilling, Plasma De-Smear Technology, Laser Microvia, Conductive Plate Shut.

Take the Accutrace Challenge and see WHY OUR PRICING CANNOT BE BEATEN

Accutrace inc. www.PCB4u.com sales@PCB4u.com

● SAM & ITAR Registered ● I SO 9001 - 2008, ● UL E33047

MINI MANTIS

$299.99

PART: 637179

The Mini Mantis™ 6WD rover will take your off-road robotics excursions to the next level!

The bug-like chassis offers extreme A-arm style suspension and with 485 rpm Economy Gear Motors, you'll have plenty of torque to turn the aggressive off-road tires even when up against the toughest obstacles. The boxed channel chassis provides a rigid backbone with endless mounting and customization options, making it easy to bolt Actobotics® components and electronics directly to the rover.

Mini Mantis Features:

- 6 Wheel Drive
- Extreme Suspension
- High Traction Tires

EXPLORE

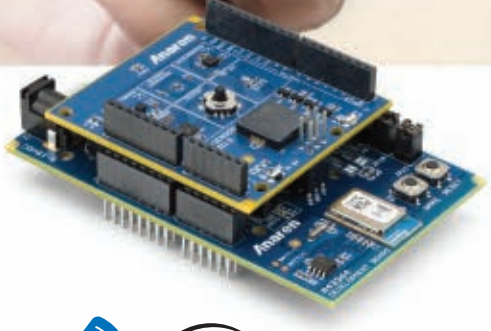
ENGAGE

EXTEND

MORE PRODUCT INFO!

ACTOBOTICS®
DREAM • DESIGN • BUILD • REPEAT

AVAILABLE AT SERVOCITY.COM
sales@servocity.com • 620-221-0123



A giant, coffee-carrying IoT dinobot at the command of your index finger? Why not. In fact, wherever your imagination leads you, Anaren empowers you with the industry's easiest-to-use wireless connectivity platform. Scan the code to learn more and register for a chance to win a free Atmosphere MSDK.* Think. Build. Connect.

AIR Anaren Integrated Radio

WICED by CYPRESS

ATMOSPHERE Featuring Anaren Stratosphere Cloud Hosting

Anaren® What'll we think of next?®

* Must be 18 to participate. No purchase necessary. North America only. Void where prohibited. Other restrictions apply.

www.anaren.com/air/win